Mondragon
Unibertsitatea

DOCTORAL THESIS

METHODOLOGY FOR THE ACCELERATED RELIABILITY ANALYSIS AND PROGNOSIS OF UNDERGROUND CABLES BASED ON FPGA

UNAI GARRO ARRAZOLA

Arrasate-Mondragón, 2019

METHODOLOGY FOR THE ACCELERATED RELIABILITY ANALYSIS AND PROGNOSIS OF UNDERGROUND CABLES BASED ON FPGA

Thesis subitted to

MONDRAGON UNIBERTSITATEA

for the degree of

DOCTOR BY MONDRAGON UNIBERTSITATEA

Submitted by

UNAI GARRO ARRAZOLA

Supervisors

DR. EÑAUT MUXIKA OLASAGASTI
DR. JOSE IGNACIO AIZPURUA UNANUE

NOVEMBER 2019

DECLARATION OF ORIGINALITY

I hereby declare, that this Thesis is my original work, which I have worked out by my own. All sources, references and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Arrasate-Mondragón, November 2019

ABSTRACT

Dependable electrical power distribution systems demand high reliability levels that cause increased maintenance costs to the utilities. Often, the extra costs are the result of unnecessary early maintenance procedures, which can be avoided by monitoring the equipment and predicting the future system evolution by means of statistical methods (prognostics).

The present thesis aims at designing accurate methods for predicting the degradation of high and medium voltage underground Cross-Linked Polyethylene (XLPE) cables within an electrical power distribution grid, and predicting their remaining useful life, in order inform maintenance procedures.

However, electric power distribution grids are large, components interact with each other, and they degrade with time and use. Solving the statistics of the predictive models of the power grids currently requires long numerical simulations that demand large computational resources and long simulation times even when using advanced parallel architectures. Often, approximate models are used in order to reduce the simulation time and the required resources.

In this context, Field Programmable Gate Arrays (FPGAs) can be employed to accelerate the simulation of these stochastic processes. However, the adaptation of the physics-based degradation models of underground cables for FPGA simulation can be complex. Accordingly, this thesis proposes an FPGA-based framework for the on-line monitoring and prognosis of underground cables based on an electro-thermal degradation model that is adapted for its accelerated simulation in the programmable logic of an FPGA.

RESUMEN

Las redes de distribución de energía eléctrica confiables requieren de altos niveles de fiabilidad, que causan un mayor coste de mantenimiento a las empresas distribuidoras. Frecuentemente los costes adicionales son el resultado de procedimientos de mantenimiento innecesarios, que se pueden evitar por medio de la monitorización de los equipos y la predicción de la evolución futura del sistema, por medio de métodos estadísticos (prognosis).

La presente tesis pretende desarrollar métodos adecuados para la predicción de la degradación futura de cables de alta y media tensión Cross-Linked Polyethylene (XLPE) soterrados, dentro de una red de distribución eléctrica, y predecir su tiempo de vida restante, para definir una secuencia de mantenimiento óptima.

Sin embargo, las redes de distribución eléctrica son grandes, y compuestas por componentes que interactúan entre sí y se degradan con el tiempo y el uso. En la actualidad, resolver estas estadísticas predictivas requieren grandes simulaciones numéricas que requieren de grandes recursos computacionales y largos tiempos de simulación, incluso utilizando arquitecturas paralelas avanzadas.

Las Field Programmable Gate Array (FPGA) pueden ser utilizadas para acelerar las simulaciones de estos procesos estocásticos, pero la adaptación de los modelos físicos de degradación de cables soterrados para su simulación en una FPGA puede ser complejo. Así, esta tesis propone el desarrollo de una metodología de monitorización y prognosis cables soterrados, basado en un modelo de degradación electro-térmico que está adaptado para su simulación acelerada en la lógica programable de una FPGA.

LABURPENA

Energia elektrikoaren banaketa-sare konfidagarriek fidagarritasun maila altuak eskatzen dituzte, eta honek beraien mantenketa kostuen igoera dakar. Kostu hauen arrazoia beraien bizitzan goizegi egiten diren mantenketa prozesuei dagokie askotan, eta hauek eragoztea posible da, ekipamenduaren monitorizazioa eginez eta sistemaren etorkizuneko eboluzioa aurrez estimatuz (prognosia).

Tesi honen helburua lurpeko tentsio altu eta ertaineko Cross-Linked Polyethylene (XLPE) kable sistemen eboluzioa eta geratzen zaien bizitza aurreikusiko duten metodo egokiak definitzea izango da, banaketa-sare elektriko baten barruan, ondoren mantenketa prozesu optimo bat ahalbidetuko duena.

Hala ere, sistema hauek oso jokaera dinamikoa daukate. Konponente ezberdinek beraien artean elkar eragiten dute eta degradatu egiten dira denboran eta erabileraren ondorioz. Estatistika hauen soluzio analitikoa lortzea ezinezkoa da gaur egun, eta errekurtso asko eskatzen dituen simulazio luzeak behar ditu zenbakizko erantzun bat lortzeko, arkitektura paralelo aurreratuak erabili arren.

Field Programmable Gate Array (FPGA)k prozesu estokastiko hauen simulazioa azkartzeko erabil daitezke, baina lurpeko kableen degradazio prozesuen modelo fisikoak FPGA exekuziorako egokitzea konplexua izan daiteke. Beraz, tesi honek FPGA baten logika programagarrian azeleratu ahal izateko egokitua izan den degradazio elektrotermiko modelo baten oinarritutako monitorizazio eta prognosi metodologia bat proposatzen du.

ACKNOWLEDGEMENTS

I have to say thanks to so many people that if I tried mentioning all of them I would probably forget too many important people that have been besides me during these last years. People that pointed me towards the right direction, people that helped me with technical software, offered me the right books and tools, or simply made me smile any day. Really thank you all!

Especial thanks to my thesis director Eñaut Muxika for kindly attending all my darkest questions, and to the co-director Joxe Aizpurua for his support through the publishing processes and the pursuing of financing. Also thanks to Mikel Mendikute for obtaining funds and helping in the review process of several manuscripts, and to Victoria Catterson and Brian Stewart for their enthusiastic ideas that pointed me in that direction at the beginning of the thesis.

Thanks also to Jose Maria Canales for his help with the understanding of actual electrical power distribution grids, to Xabier Artetxe for his support on the validation of certain mathematical proofs, and to Iban Barrutia for his support and encouragement through the thesis.

My appreciation to all those that spent part of their personal time and efforts reviewing this document.

Thanks also to the University of Illinois and University of Catania for kindly providing their reliability analysis software Möbius Tool.

And finally, to the faculty and colleagues of MGEP for offering me the great opportunity of going through this thesis project.

CONTENTS

I APPENDIX

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AC	Alternating Current
AGAN	As Good as New
ARNN	Adaptive Recurrent Neural Networks
ASIC	Application-Specific Integrated Circuit
ATS	Adaptive Transition System
AXI	Advanced Extensible Interface
BDMP	Boolean Driven Markov Process
BRAM	Block RAM
CB	Circuit Breaker
CBM	Condition Based Maintenance
CDF	Cumulative Distribution Function
CLB	Configurable Logic Block
CM	Condition Monitoring
CPU	Central Processing Unit
CSAN	Coloured Stochastic Activity Network
CTMC	Continuous Time Markov Chain
DFT	Dynamic Fault Tree
DMA	Direct Memory Access
DSP	Digital Signal Processor
DTR	Dynamic Thermal Rating
DWNN	Dynamic Wavelet Neural Networks
EAB	Elongation At Break
EOL	End Of Life
FEM	Finite Element Method
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FSPN	Fluid Stochastic Petri Net
FT	Fault Tree
GBDMP	Generalized Boolean Driven Markov Process
GDT_SPN	Generally Distributed Transition Stochastic Petri Net
GFT	Generalized Fault Tree
GPU	Graphics Processing Unit
GSMP	Generalized Semi-Markov Process
GSPN	Generalized Stochastic Petri Net
HDL	Hardware Description Language
HLS	High-Level Synthesis
HPOF	High-Pressure Oil-Filled

HSAN	Hierarchical Stochastic Activity Network
HVAC	High Voltage Alternative Current
HW	Hardware
ICDF	Inverse Cumulative Distribution Function
IIR	Infinite Impulse Response
IP	Intellectual Property
IPM	Inverse Power Model
LFSR	Linear Feedback Shift Register
LTI	Linear Time-Invariant
LUT	Lookup Table
MAC	Macro Activity Class
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
NO	Normally open
OCM	On-Chip Memory
PD	Partial Discharge
PDF	Probability Density Function
PDMP	Piecewise-Deterministic Markov Process
PF	Particle Filter
PFT	Parametric Fault Tree
PL	Programmable Logic
PN	Petri Net
PRNG	Pseudo-Random Number Generation
RAM	Random Access Memory
RBD	Reliability Block Diagram
RFT	Repairable Fault Tree
RUL	Remaining Useful Life
SAN	Stochastic Activity Network
SCOF	Self-Contained Oil-Filled
SFT	Static Fault Tree
SMC	Sequential Monte Carlo
SPN	Stochastic Petri Net
SW	Software
TCL	Tool Command Language
TFT	Temporal Fault Tree
TMR	Triple Modular Redundancy
TOF	Time of Failure
TTF	Time To Failure
UKF	Unscented Kalman Filters
URN	Uniform Random Number
VEC	Voltage Endurance Coefficient
VFPU	Vector Floating-Point Unit

VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
XLPE	Cross-Linked Polyethylene
XML	EXtensible Markup Language

LIST OF SYMBOLS

List of Symbols: Generic Mathematical Notation

$\mathrm{Arg}(\cdot)$	Argument
γ	Euler–Mascheroni constant $\lim_{n\to\infty}\left(-\ln n+\sum_{k=1}^{n}\frac{1}{k}\right) = 0.5772156649\ldots$
$\mathrm{Ei}\,(\cdot)$	Exponential integral function
$\mathrm{E}_n\,(\cdot)$	Generalized exponential integral function for complex operands
$\mathrm{E}_1\,(\cdot)$	Exponential integral function for complex operands
$\mathcal{L}\{\cdot\}$	Laplace transform
$\mathcal{P}\,(\cdot)$	Probability
s	Complex frequency variable used in a Laplace transform
t	time (s)

List of Symbols: Signal Processing

$\delta[n]$	Discrete unit impulse
$\delta(t)$	Dirac delta signal
n	Discrete-time index
τ_s	Discrete sampling time interval (s)
$v(t)$	Unit step signal
ω_i	FIR filter weight number i
$x[n]$	Discrete sample of signal x at time index n

List of Symbols: SAN models

A	Activity set of a SAN activity network
ac_i	Activity number i
c	Case list set of a SAN activity network
$\mathrm{Ct}(s)$	Completion time set of SAN model s
$\mathrm{Ena}_{ac}(ac_i)$	Enablement status of activity ac_i
$f_g(ig_i)$	Function associated to input gate ig_i
$f_g(og_j)$	Function associated to output gate og_j
ι	Activity to input gate map set of a SAN activity network
I	Input gate set of a SAN activity network

$\mathrm{IG}(ac_i)$	Input gate set of activity ac_i
$\mathrm{m}(pl_k)$	Marking of place pl_k
μ_{SAN}	Marking set of SAN model
N_{ac}	Number of activities in a SAN model
og_i	Output gate number i
o	Activity to output gate map set of a SAN activity network
O	Output gate set of a SAN activity network
$\mathrm{OG}(ac_i)$	Output gate set of activity ac_i
pl_i	Place number i
P	Place set of a SAN activity network
$Pl(g)$	Place set associated to gate g
$\mathrm{Pred}_{ig}(ig_i)$	Predicate function associated to input gate ig_i
t_{ac_i}	Absolute completion time of activity ac_i (measured from the beginning of the simulation) (s)
τ_{ac_i}	Completion time of activity ac_i (measured from the time that it is enabled) (s)
ν	Activity type set of a SAN activity network

List of Symbols: Power Distribution Grids

cbl	Cable
cb	Circuit breaker
f_{Load}	Load factor
tf	Transformer

List of Symbols: Reliability and Probability

CI	Confidence interval
Exp	Exponential distribution
F_X	Cumulative distribution function of random number sequence X
f_F	Probability density function of the system or equipment failure
F	Faulted
$\overline{F}$	Not faulted
$F_F(t)$	Unreliability of the system at time t
F_X^{-1}	Inverse cumulative distribution function of random number sequence X
f_X	Probability density function of random number sequence X
$\mathcal{P}(pl_i, t)$	Probability that place pl_i is marked with a marking of value greater than zero at time t

Q	Unavailability
$\hat{Q}$	Estimated value of unavailability
$Q(t)$	Unavailability of the system at time t
$\mathcal{R}(t)$	Reliability of the system at time t
$\mathcal{R}_{\text{min}}$	Minimum acceptable reliability value for the system or equipment
t_{F}	Time of failure of the system or equipment (s)
U	Uniform distribution
Weib	Weibull distribution

LIST OF SYMBOLS: MONTE CARLO SIMULATION

$CI(n_{cl})$	Confidence interval with confidence level n_{cl}
$\text{m}(pl_j, i, t)$	Marking of place pl_j at time instant t of the MC trial number i
$\mu_i^{(j)}(t)$	Marking of place pl_i at time instant t of the MC trial number j
n_{cl}	Confidence level of the given confidence interval (%)
N_{MC}	Number of MC trials performed
t_{SIM}	Simulation time instant (s)
S	Square root of the sample variance or standard deviation of the samples for the samples of a Monte Carlo (MC) simulation at a given time
z_x	x quantile of the standard normal distribution

LIST OF SYMBOLS: FPGA SAN SIMULATOR ARCHITECTURE

ac_{comp}	Completed activity.
ac_{Phys}	Activity driven by a physical model.
bk_i	FPGA block number i
$G_{ac\,\text{att}}$	Gain obtained in the speed or reduction of the clock cycles required to attend all activities, in comparison to a sequential search.
$\text{ltn}(bk_i)$	Latency of block bk_i in clock cycles (clock cycles)
$\text{ltn}_{\mathcal{M}}$	Latency of the minimum completion time and activity index algorithm (clock cycles)
$n_{ac\,\text{req}}$	Average number of activity requests after each activity completion
$\Psi_o(X_M)$	Priority encoder function with M request input set X_M at offset o
τ_{SIM}	Time required to perform the simulation (s)
t_M	Mission time (s)
$t_{M,\text{Phys}}$	Mission time limit issued to a physical model (s)

$\tau_{ac\text{att}}$	Number of clock cycles required to attend all activities per activity completion (clock cycles)
$\tau_{\text{act}}(ac_k, t_{\text{SIM}})$	Number of clock cycles required to complete ac_k at simulation time t_{SIM} (clock cycles)
τ_{coll}	Number of clock cycles required to collect the SAN activity states (clock cycles)
$\tau_{\text{IG}}(ac_k, t_{\text{SIM}})$	Set of number of cycles required to execute the input gates associated with activity ac_k at simulation time t_{SIM}
$\tau_{\text{OG}}(ac_k, t_{\text{SIM}})$	Set of number of cycles required to execute the output gates associated with activity ac_k at simulation time t_{SIM}
τ_{pip}	Total latency of the pipeline (clock cycles)
τ_{samp}	Sample latency in a pipeline (clock cycles)
$\tau_{\text{Sch}}(t_{\text{SIM}})$	Latency of the scheduler at simulation time t_{SIM} (clock cycles)
$\tau^{(ac_i)}_{\text{trig},ac_j}(t_i)$	Number of clock cycles required to trigger and complete activity ac_j after the completion of ac_i, at simulation time $t_{\text{SIM}} = t_i$ (clock cycles)

List of Symbols: Electro-Thermal Model

U_0	Energy barrier for the probability of breaking the bonds of a material in a Zhurkov model (J)
α_{c}	Temperature coefficient of the electrical resistivity of the main conductor (K^{-1})
β_E	Shape parameter of the Weibull distribution that describes the electrical field strength required for the insulator breakdown
c	Volumetric thermal capacity of a material ($\text{J}\,\text{m}^{-3}\,\text{K}^{-1}$)
C_{a}	Thermal capacitance per unit length of the cable armor ($\text{J}\,\text{m}^{-1}\,\text{K}^{-1}$)
C_{c}	Thermal capacitance per unit length of the main conductor ($\text{J}\,\text{m}^{-1}\,\text{K}^{-1}$)
C_{e}	Thermal capacitance per unit length of the external cover of the cable ($\text{J}\,\text{m}^{-1}\,\text{K}^{-1}$)
χ	Structural parameter of the material in a Zhurkov degradation model ($\text{J}\,\text{m}\,\text{mol}^{-1}\,\text{V}^{-1}$)
C_i	Thermal capacitance per unit length of cable insulation ($\text{J}\,\text{m}^{-1}\,\text{K}^{-1}$)
c_i	Volumetric thermal capacity of the cable layer i ($\text{J}\,\text{m}^{-3}\,\text{K}^{-1}$)
C'_i	First segment of the equivalent Van Wormer capacitance set of C_i
C''_i	Second segment of the equivalent Van Wormer capacitance set of C_i ($\text{J}\,\text{m}^{-1}\,\text{K}^{-1}$)

$C_{\mathrm{i}x}$	Thermal capacitance per unit length of cable insulation layer x ($\mathrm{J\,m^{-1}\,K^{-1}}$)
c_{s}	Volumetric thermal capacity of the soil ($\mathrm{J\,m^{-3}\,K^{-1}}$)
C_{sh}	Thermal capacitance per unit length of the cable sheath ($\mathrm{J\,m^{-1}\,K^{-1}}$)
$\mathcal{D}$	Enlargement factor that relates full sized cable insulator and laboratory sample insulator lifetimes
D_{e}	External diameter of cable (jacket/cover) (m)
δ_{s}	Thermal diffusivity of the soil ($\mathrm{m^2\,s^{-1}}$)
d_{pk}	Distance between cable p and cable k (m)
d'_{pk}	Distance between cable p and the mirror above the ground of cable k following the Kenelly hypothesis (m)
d_{s}	Dry density of the soil ($\mathrm{kg\,m^{-3}}$)
$\Delta W[n]$	Increment in the dissipated power at discrete time index n ($\mathrm{W\,m^{-1}}$)
$\Delta\xi_i$	Incremental degradation caused by the i^{th} stress level
E	Electric field strength ($\mathrm{V\,m^{-1}}$)
E_0	Electric field strength threshold value at which the degradation of a polymer is considered negligible ($\mathrm{V\,m^{-1}}$)
$E_{d63,x}$	63% quantile of the Weibull distribution describing the electrical field strength required for the insulator breakdown of cable x
I_0	Rated current of the cable (A)
k_B	Boltzmann constant $1.380\,649 \times 10^{-23}\,\mathrm{J\,K^{-1}}$
λ_{a}	Loss factor of a cable armor
λ'_{a}	Loss factor of a cable armor due to circulating currents
λ''_{a}	Loss factor of a cable armor due to hysteresis in a magnetic armor currents
$\lambda_{\mathrm{a,mid}}$	Loss factor of the armor of the cable in the center of a cable-set
$\lambda_{\mathrm{a,side}k}$	Loss factor of the armor of the side cable k in a cable-set
λ_{sh}	Loss factor of a cable sheath
λ'_{sh}	Loss factor of a cable sheath due to circulating currents
λ'_{sh}	Loss factor of a cable sheath due to circulating currents
λ''_{sh}	Loss factor of a cable sheath due to the eddy currents
$\lambda_{\mathrm{a,mid}}$	Loss factor of the sheath of the cable in the center of a cable-set
$\lambda_{\mathrm{sh,side}k}$	Loss factor of the sheath of the side cable k in a cable-set
L_d	Depth of burial of the cable (m)

L_θ	Number of discrete-time samples required for the thermal process to approximate a first order thermal respose
M_θ	Number of discrete-time samples to the inflection point on the thermal step response of the mutual heating process or the soil effect process, where the thermal process response decelerates exponentially
$M_{\theta e}$	Number of discrete-time samples to the inflection point on the thermal step response of the soil effect process, where the thermal process response decelerates exponentially
$M_{\theta pk}$	Number of discrete-time samples to the inflection point on the thermal step response of the mutual heating process, where the thermal process response decelerates exponentially
N_φ	Number of phases of a cable set
ϕ_s	Moisture content of the soil as % of the dry density (%)
p_i	Van Wormer factor for insulator layer i
p_i^*	Van Wormer factor for the sub-segments of insulator layer i
Q	Total transferred energy in a heat transfer process (J)
r	Radius (m)
R_G	Universal gas constant $8.314\,462\,618\,153\,24\,\mathrm{J\,K^{-1}\,mol^{-1}}$
ϱ	Thermal resistivity ($\mathrm{K\,m\,W^{-1}}$)
ϱ_i	Thermal resistivity of cable layer i ($\mathrm{K\,m\,W^{-1}}$)
ϱ_s	Thermal resistivity of the soil ($\mathrm{K\,m\,W^{-1}}$)
ρ_x	Electrical resistivity of a conductor x ($\Omega\,\mathrm{m}$)
$\rho_{x,\theta_\mathrm{ref}}$	Electrical resisitivity of a conductor x at reference temperature θ_ref ($\Omega\,\mathrm{m}$)
r_i	Outer radius of cable layer i (m)
$r_{i,\mathrm{in}}$	Inner radius of cable layer i (m)
$r_{i,\mathrm{out}}$	Outer radius of cable layer i. Same as r_i, used for clarity when $r_{i,\mathrm{in}}$ is used in the same context (m)
S	Surface area ($\mathrm{m^2}$)
τ_0	Remaining lifetime under reference or no stress conditions (s)
τ_RUL	Remaining useful lifetime (s)
τ_F	Time to failure or remaining lifetime (s)
θ	Temperature (°C)
ϑ	Absolute temperature (K)
θ_c	Temperature of the main conductor (°C)
$\theta_e(t)$	Temperature rise caused due to the cable being buried underground (°C)
θ_in	Inner temperature (°C)

$\theta_M(t)$	Temperature of the main conductor rise calculated for time t, considering the resistivity in steady state conditions (°C)
θ_{out}	Outer temperature (°C)
$\theta_{pdk}(t)$	Temperature rise caused on cable p by the dielectric losses in a neighboring cable k (°C)
$\theta_{pk}(t)$	Temperature rise caused on cable p by the power dissipated in a neighboring cable k, or mutual heating (°C)
θ_{ref}	Reference temperature (°C)
$\theta_s(t)$	Temperature rise caused by the total power dissipated in the cable, or self heating (°C)
$\theta_\delta(t)$	Impulse response of a thermal process (°C m W^{-1})
$\theta_{\delta x}(t)$	Impulse response of a thermal process x (°C m W^{-1})
$\theta_v(t)$	Unit step response of a thermal process (°C m W^{-1})
$\theta_{vx}(t)$	Unit step response of thermal process x (°C m W^{-1})
$\theta_{Wv}(t)$	Response of a thermal process to a constant dissipated power step of value W (°C)
$\theta_M(\infty)$	Temperature of the main conductor in steady state, considering the increased conductor resistivity due to temperature (°C)
$\mathcal{T}_i$	Thermal resistance per unit length of cable layer i (K m W^{-1})
V_0	Rated voltage of the cable (A)
V_i	Volume per unit length or cross-section of insulator layer i (m^2)
W	Linear density of a constant dissipated power (W m^{-1})
w_0	Activation energy in an Zhurkov degradation model (J mol^{-1})
W_{a}	Linear density of the power dissipated in the cable armor (W m^{-1})
U_{Arr}	Activation energy in an Arrhenius degradation model (J)
W_{c}	Linear density of the power dissipated in the main conductor (W m^{-1})
W_{e}	Linear density of the power dissipated in the cable that is conducted to the exterior (W m^{-1})
W_{sh}	Linear density of the power dissipated in the cable sheath (W m^{-1})
$W(t)$	Time-varying linear density of the power dissipated in a cable, due to a time-varying load (W m^{-1})
W_{tot}	Linear density of the total power dissipated within a cable (W m^{-1})
$W_{\mathrm{tot},s}$	Linear density of the total power dissipated within the cable under analysis (W m^{-1})

$W_{\mathrm{tot,side}}$	Linear density of the total power dissipated within a side cable ($\mathrm{W\,m^{-1}}$)
ξ	Cumulated degradation or life loss
$\zeta(t)$	Attainment factor

List of Symbols: Adapted Electro-Thermal Model

$\Delta\tau_{\mathrm{RUL}}$	Deviation of the system RUL due to the uncertainties of the system (s)
ΔW_{c}	Power density transient applied to the main conductor of a cable ($\mathrm{W\,m^{-1}}$)
$\Delta\hat{W}_{\mathrm{c}}$	Estimated power density transient applied to the main conductor of a cable ($\mathrm{W\,m^{-1}}$)
$\Delta\hat{\theta}(t)$	Estimated value of the temperature transient in a cable caused by a constant power transient (°C)
$\Delta\theta(t)$	Temperature transient in a cable caused by a constant power transient (°C)
$\Delta\xi(t)$	Degradation transient caused by a constant power transient on the main conductor of a cable
$\varepsilon_{\Delta W_{\mathrm{c}}}(t)$	Error in the estimation of the power density transient applied ot the main conductor of a cable ($\mathrm{W\,m^{-1}}$)
$\varepsilon_{\Delta\theta}(t)$	Error in the estimation of a temperature transient caused by a constant power transient (°C)
$\varepsilon_{\Delta\xi}(t)$	Error in the estimation of the degradation transient caused by a constant power transient on the main conductor of a cable
$\varepsilon_{I}(t)$	Error in the estimation of the current applied to a cable (A)
$\varepsilon_{\theta}(t)$	Error in the estimation of the temperature of the main conductor of a cable (°C)
$I(t)$	Current applied to a cable (A)
R	Electrical resistance of a conductor (Ω)
t_0	Time instant when the prognostics analysis takes place (s)
τ_c	Time constant of the equivalent IIR filter on the last stage of the mutual heating and soil effect process implementations (s)
τ_e	Time constant of the equivalent IIR filter on the last stage of the soil effect process implementation (s)
τ_{pk}	Time constant of the equivalent IIR filter on the last stage of the mutual heating process implementation (s)
$\theta_e[n]$	Discrete-time temperature rise caused by the effect of the cable being buried in the soil, at discrete-time instant n (°C)

$\theta_{pk}[n]$	Discrete-time temperature rise caused by the power dissipated in a neighboring cable k on cable p, or mutual heating, at discrete-time instant n ($^\circ\mathrm{C}$)
$\hat{\theta}_{v}^{(j)}$	Approximate weight coefficient for the j^{th} segment filter for either the mutual heating or soil effect processes
$\hat{\theta}_{v,e}^{(j)}$	Approximate weight coefficient for the j^{th} segment filter for the soil effect process
$\hat{\theta}_{v,pk}^{(j)}$	Approximate weight coefficient for the j^{th} segment filter for the mutual heating process
$\theta_{v,e}[n]$	Discrete-time thermal unit step response of the soil effect ($^\circ\mathrm{C\,m\,W^{-1}}$)
$\theta_{vx}[n]$	Discrete unit step response of thermal process x ($^\circ\mathrm{C\,m\,W^{-1}}$)
$\theta_{v}[n]$	Discrete unit step response of a thermal process ($^\circ\mathrm{C\,m\,W^{-1}}$)
$\theta_{v,pk}[n]$	Discrete-time thermal unit step response of the mutual heating process ($^\circ\mathrm{C\,m\,W^{-1}}$)
$\theta_{v,P_l,L_l}[n]$	Segment of the discrete unit step response of a thermal process beginning at time index $\theta_v[P_l]$ with length L_l ($^\circ\mathrm{C\,m\,W^{-1}}$)
$\xi_{\lim}$	Maximum acceptable limit of the cable degradation level $\xi(t)$ for an adequate power delivery

GLOSSARY

Activity	A primitive of the Stochastic Activity Network (SAN) model that defines the stochastic or deterministic time intervals that govern the model (Movaghar, 1984).
Activity completion	An activity in SAN model defines a stochastic/deterministic/instantaneous time interval, after which the activity is said to *complete* by executing the functions defined in its associated input gates and output gates (Movaghar, 1984).
Ampacity	The maximum current or current rating that can be applied to a cable in a given installation conditions, without exceeding its temperature rating (Wild et al., 2015).
Armor	Protective metallic layer of a cable made of steel wires or tapes (G. J. Anders, 2005, Chapter 1).
As-good-as-new	(AGAN): Condition of an equipment after a repair that restored the equipment to the initial conditions (Nikulin et al., 2010).
Clock enable	(`CE`): Signal in an Field Programmable Gate Array (FPGA) implementation that enables or disables the clock signal to part of the blocks or parts of a pipeline, in order to stall its execution.
Condition monitoring	(CM): detection and collection of information and data that indicate the state of a machine (ISO, 2012).
Conductor screen	See shield.
Data hazard	Situation on a pipelined processor where multiple stages in a pipeline depend from each other, and the input required by one of the stages is unavailable until the output of another stage finishes its processing (Omondi, 1999, Chapter 5).
Diagnostics	Examination of symptoms and syndromes to determine the nature of faults or failures (kind, situation, extent) (ISO, 2012).
Failure events	Events that describe occurrences that cause a functioning system to enter a failed state.
Fault events	Events that describe occurrences that cause a functioning component to enter a failed or faulted state. The process may or may not lead ot a failed system state.
Flush	Process of invalidating part of the pipeline because of invalid input data(Omondi, 1999, Chapter 4).
Hazard	Condition that can cause injury or death, damage to or loss of equipment or property, or environmental harm (Bahr, 2014).

Input gate	A primitive of the SAN model that defines the marking conditions that enable an activity completion and the functions that alter the associated place markings upon the activity completion (Movaghar, 1984).
Load factor	The ratio of the average power demand to the peak demand in a power distribution grid (Gönen, 2014).
Operational events	Events that describe occurrences that are part of the normal non-failed behavior of a system.
Output gate	A primitive of the SAN model that defines the functions that alter the associated place markings upon the associated activity completion (Movaghar, 1984).
Place	A primitive of the SAN model that holds a marking that defines the state of the model (Movaghar, 1984).
Priority encoder	Combinational circuit that outputs the index of the block input with highest priority among those inputs that are enabled.
Prognosis	Process that permits obtaining the prognostics of certain equipment.
Prognostics	The capability to provide early detection of the precursor and/or incipient fault condition (very *"small"* fault) of a component, and to have the technology and means to manage and predict the progression of this fault condition to component failure (Engel et al., 2000), or the process of predicting the future reliability of a product by assessing the extent of deviation or degradation of the product from its expected normal operating conditions (Pecht, 2008).
Reliability	$\mathcal{R}(t)$: The probability that the component will not experience any failure during the time interval $[0,t]$, given that the component was repaired at time zero.
Remaining useful life	(RUL): Time difference between the current time, up to the time that the system or equipment reaches a condition that it no more performs its intended function (Zio, 2016). This time limit is usually dictated by the time a maximum acceptable degradation level is reached, or the reliability of the equipment crosses below the reliability requirement threshold.
Risk	The combination of the probability of occurrence and consequence (or severity) of a hazard (Bahr, 2014).
Screen	Semiconducting (not to be confused with semiconductor materials) layer between the insulator and the metallic sheath of a cable, for the purpose of equalizing the electric field strength (Ametani, Ohno, and Nagaoka, 2015).
Sheath	Metallic layer of a cable, provided for the purpose of insulating the cable from moisture (Hyvönen et al., 2008).

Shield	Also called conductor screen, it is a semiconducting (not to be confused with semiconductor materials) layer between the main conductor and the insulator of a cable, for the purpose of equalizing the electric field strength (Ametani, Ohno, and Nagaoka, 2015, Chapter 1).
Shift register	Sequential circuit that stores a given input data and shifts it toward the output at every clock cycle (Bazil Raj, 2018).
Stall	Paused state that a pipelined architecture enters when a data hazard occurs (Omondi, 1999, Chapter 5).
Stateful	FPGA block element characteristic where the current output depends on a variable that represents the current state of the system.
Stateless	FPGA block element characteristic where the block has no notion of state, that is, the current output depends only on the given inputs.
Synthesis	The process by which the design entry for an FPGA is converted into a specific implementation at gate level (Seals and Whapshott, 1997).
Thermal rating	The process to calculate the maximum current that can be applied to an equipment, without causing critical damage.
Throughput	Number of output samples per unit of time produced by an FPGA block.
Unavailability	$Q(t)$: The probability that the component is in failed state at time t (Kumamoto and Henley, 1996).
Unreliability	$F_F(t)$: The probability that the component experiences the first failure during the time interval $[0, t)$, given that the component was repaired at time zero (Kumamoto and Henley, 1996).

1

INTRODUCTION

The power generation industry is facing a demand to increase the reliability of their networks and reduce their costs at the same time. Maintaining the reliability of equipment can require implementing maintenance procedures that result in either replacing or repairing the electrical grid equipment. These maintenance procedures increase the costs of electric power distribution, and they can be the limiting factor for the useful lifetime of the equipment (*economical lifetime* (Gönen, 2014)). Therefore, the reliability and the cost reduction may seem to be contradictory.

However, failures in the electricity distribution service can lead to losses and fire hazards can put people at risk (Chakravorti, Dey, and Chatterjee, 2013). In this context, it is compulsory to define appropriate maintenance procedures that reduce the risk, and they can lead to the reduction of costs due to downtimes.

It is estimated that 80% of the outages in the service to the customer are caused by electricity distribution equipment failures (Billinton and W. Li, 1994). This highlights the need to maintain each individual equipment operative. However, a systematic periodical maintenance procedure can be unacceptable due to high costs. In this context, the total maintenance costs can be reduced by implementing Condition Based Maintenance (CBM) procedures that diagnose the equipment condition and plan the maintenance procedures accordingly (Coble et al., 2012). However, planning the maintenance procedures solely based on the current condition of the equipment can be risky and costly in dependable distribution grids. Dependable power distribution systems require that the probability of the outages, fires or any other hazard is minimized, and accordingly, the probability of these events needs to be evaluated.

Costs and risks of the system can be reduced if adequate procedures, called prognostics are adopted to predict the evolution of the degradation level of the system or equipment. The procedures should estimate the time when the probability of failure of the system will be unacceptable. Maintenance procedures will need to adopted before this time, so as to maintain the reliability of the system above the required limit.

In this context, an adequate planning of maintenance procedures based on prognostics of the grid system can reduce the maintenance costs and increase the reliability of the distribution system. The analysis results would permit reducing, grouping, scheduling and optimizing the costs of maintenance procedures, while guaranteeing the required reliability levels for the grid.

However, the analysis of a full electric power distribution grid can be complex. Structures are composed of thousands of elements that degrade with the use and weather conditions and change states during their service time. The customers also vary the power demand and accordingly this causes changes in the system to vary their service accordingly. Customer usage patterns follow complex statistics (Billinton and W. Li, 1994). Therefore, the grids are dynamic systems that suffer varying degradation rates from varying load levels.

The failure mechanism of a grid is also very intricate. The failure of certain components in the grid can cause a new condition on other components of the grid that varies the rates at which these other components degrade. This sequence can lead to the full system failure. Therefore, the prognostics of a given component cannot be performed in isolation. Instead, it requires taking into account the influence of the grid on the component and vice versa.

The prognosis of certain grid equipment such as power transformers has been thoroughly analyzed due to their high cost (Catterson, 2014; J. I. Aizpurua, McArthur, et al., 2019; Djeziri, Benmoussa, and Benbouzid, 2019). However, the degradation models, condition monitoring and prognosis of other equipment such as Circuit Breakers (CBs) are still under active research due to the complexity and diversity of the component structure (Feizifar and Usta, 2019; Zhong et al., 2015; Hosseini et al., 2018; J. I. Aizpurua, Catterson, Abdulhadi, et al., 2018).

The prognostics analysis of underground cables has been paid little attention until recently. Their long Time To Failure (TTF) in comparison with other components has possibly reduced the interest on this component in the past. However, the prediction of the failure time of a cable plays a crucial role in the reliability analysis of an electric power distribution grid (Shumaker et al., 2013), and the costs of manufacturing and replacement of underground cables can be costly (Robles et al., 2016; Aras, Alekperov, et al., 2007).

Monitoring the cable lifetime involves performing the estimation of the health state of the degradation state of the cable insulator (diagnostics), and the prediction of the evolution of the insulator wear through the cable lifetime (prognostics). While several oil-paper insulated cables are still under use since 1940, Cross-Linked Polyethylene (XLPE) cables began to be installed by the end of the 1960s, and currently typically installed land cables are XLPE, Self-Contained Oil-Filled (SCOF) or High-Pressure Oil-Filled (HPOF). Recently XLPE cables (see Figure 1.1) are replacing both SCOF and HPOF cables due to their lack of pressure system, reduced ecological impact and reduced maintenance (Hyvönen et al., 2008; Ametani, Ohno, and Nagaoka, 2015).

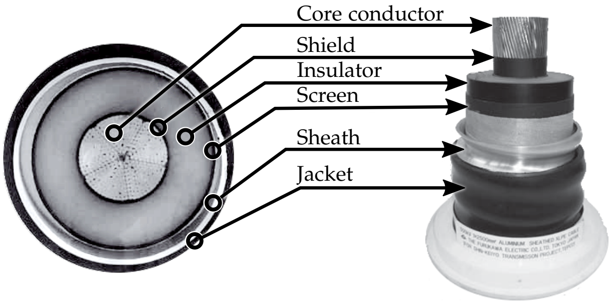


Figure 1.1: XLPE cable structure (Ametani, Ohno, and Nagaoka, 2015).

In order to estimate the Remaining Useful Life (RUL) of the insulator, it is necessary to estimate the electro-thermal degradation that the insulators will suffer in the future, given the grid conditions. The process involves evaluating the probabilities of each grid state and estimating the degradation of the polymers under such environmental and grid conditions. This process must take into account the dependency between the two processes, because a cable failure also causes changes in the state of the grid. The evaluation of the physics of electro-thermal cable degradation and the grid statistics is usually performed with large Monte Carlo (MC) simulations, because they are very complex to be solved analytically. However, MC simulations make use of excessive resources, and they are time consuming. Accordingly, approximations need to be performed (Kopsidas and S. Liu, 2018).

Field Programmable Gate Array (FPGA) architectures can be tailored to perform computations when an accurate model of the system can be adapted to them. Additionally, FPGAs can help accelerating the diagnostics and prognosis of the equipment on-line, onsite and at much reduced costs in comparison to traditional computing clusters (Aliee and Zarandi, 2011).

Accordingly, the main objective of this thesis is the development of an FPGA based architecture, which can perform the monitoring and prognosis predictions of underground cable systems, and the validation of its performance and results.

1.1 Objectives

In this context, the main objective of this thesis is performing the on-line Condition Monitoring (CM), reliability and prognostics analysis of high and medium voltage XLPE underground cables, considering the grid structure and its stochastic behavior, the degradation of components and varying power demands.

In order to meet these objectives, an FPGA-based solution is proposed. The architecture produces a numerical solution for the reliability values of the cable insulator, by performing a probabilistic analysis of the the cable degradation while it is subjected to grid load changes. The results should permit performing CBM before the cable failure event occurs or its reliability levels are below acceptable limits.

In order to obtain realistic results from a prognostics analysis, the developed architecture must:

1. Be capable of evaluating the current condition of the cable insulator based on data captured in the past.
2. Characterize the future stochastic behavior of the grid that governs the cable load, affecting its lifetime.
3. Be capable of evaluating the probable future condition of the cable insulator (prognosis), considering non-stationary loads and the expected future grid behavior.
4. Evaluate the future cable reliability values.
5. Perform the analysis on low-cost on-site equipment within reasonable computing times.

1.2 Considerations and Scope of This Research

Modeling a whole electrical power distribution grid can be a complex task due to the numerous variables affecting its behavior. Grids are composed of hundreds of components from multiple manufacturers with different structures, and cables themselves can have uneven layer thicknesses or they can be laid underground with uneven soil characteristics. In addition, varying weather conditions, terrain humidity levels, uncertainty of the grid power consumption, or maintenance procedures are complex stochastic processes that need to be modeled on their own. The detailed consideration of all those factors would be an extremely difficult task and assumptions need to be adopted to simplify their evaluation.

Some of the components in the grid can have complex structures and their accurate physical modeling would need Finite Element Method (FEM) models or complex state diagrams to represent all their possible states. Frequently, their event occurrences can be represented by equivalent or approximate stochastic models, such as Markov chains (Choonhapran and Balzer, 2007).

The research of this thesis concentrates on the RUL prediction underground cables subjected to load variations caused by a grid. A methodology is proposed for the integration of other grid components in this model. However, the prognosis of other grid components has not been addressed and it is proposed as future research work so as to obtain a more accurate grid model.

The physical structure and degradation mechanism of underground and aerial cables differ greatly. Typically aerial cables are not insulated, as opposed to underground cables, and they follow different degradation principles with parameters such as solar radiation or wind speed, that are not dominant degrading factors in underground cables (Alwan et al., 2016). Accordingly, this thesis concentrates on the analysis of underground cables. Oil-paper insulated cables were the norm for high-voltage underground cables in the past, but nowadays XLPE cables are the most common cable system (Hyvönen et al., 2008; Ametani, Ohno, and Nagaoka, 2015). Therefore, this thesis studies methods to monitor and analyze the reliability of medium and high voltage underground XLPE cables.

Nevertheless, the proposed models, architectures and methods are designed to be flexible to cover the different configurations of the electrical power distribution grid under analysis.

The components that will be considered in this thesis are:

1. Multi-phase underground XLPE cable-set structures.
2. Stochastic yearly load patterns.
3. Cables with uniform thermal and electrical characteristics along the cable, and laid at constant depth L_d under the soil surface (see also Figure 1.2).
4. Cables with coverings that insulate from soil humidity.
5. The depth of the cable L_d is considered sufficient to consider soil temperatures uniformly distributed along the cable sections under analysis (see Section 3.3.1.4).
6. Cable length l_{cable} is considered infinite in comparison to its diameter when solving the Fourier diffusion equation (see Section 3.3.1).

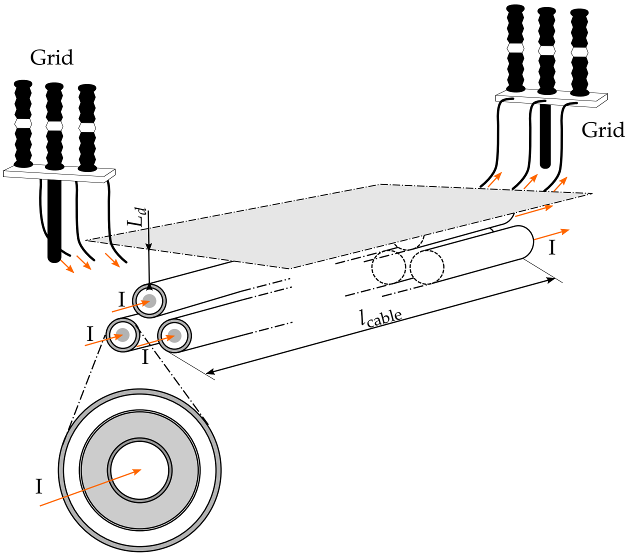


Figure 1.2: Cable-set structures considered.

The framework described through this thesis is tested with real power demand patterns obtained from (Red Eléctrica de España, 2019). The architecture is also capable of including full-year soil temperature conditions. However at the present no actual data or models have been found to be available for implementation. Therefore, in order to evaluate the impact of the methods described, all simulations have been performed under uniform soil temperature conditions.

In the models developed throughout this thesis, dimensions and thermoelectric characteristics of a cable under analysis are considered to be uniform along the section length, and enclosure, cable dimensions and thermal parameters are considered constant through the whole year. However, the possible adaptations of the models to consider their effects are discussed in Section 6.4.

The architecture permits the analysis of the impact of uncertainties on the prognosis predictions of the cable. Sensitivity analyses have been performed to evaluate this impact and to validate the platform, considering the following sources of uncertainty:

1. Predicted power demand pattern.
2. Stochastic grid switching.
3. Load sensor calibration offset during the diagnostics phase.
4. Load sensor noise during the diagnostics phase.

The former two take into account the impact of uncertainties caused by the predictions of the future load demands on the cable during the prognosis phase. The latter two consider the uncertainties during the diagnosis phase, where measurement errors can impact the evaluation of the instantaneous cable condition before the prognosis. All uncertainties except the sensor calibration offset are varying throughout each day and season of the year, while the calibration errors impact evenly through the life of a cable.

Therefore, uncertainties are considered, not only for the prediction of the future cable degradation, but also due to the measurement errors that can effect the instantaneous cable degradation estimation during its CM.

1.3 STATE OF THE ART

The estimation of reliability values for an underground cable requires performing the probabilistic analysis of:

- The cable degradation process;
- The grid model;
- Interaction between the grid and the cable component.

Accordingly, the reliability analysis of the equipment needs to be capable of modeling and simulating these structures.

Section 1.3.1 details the state of the art for the existing methods for the analysis of the degradation state of the underground cables. Sections 1.3.2 and 1.3.3 summarize the methods and formalisms existing for the modeling of electric power distribution grids and their components, Section 1.3.4 summarizes existing methods for the reliability analysis based on FPGA and similar architectures.

1.3.1 *Degradation model of the cable*

One of the main degradation factors of cables has been known to be humidity (Sanniyati et al., 2016; Takahashi, Takahashi, and Okamoto, 2008; Hyvönen et al., 2008; Montanari, Mazzanti, and Simoni, 2002; Da Silva et al., 1998). Humidity, voids and impurities are known to cause water trees when subjected to alternating electrical fields in the insulator. Physical models for the effects of the humidity in oil-paper insulated cables have been studied in the past (Hyvönen et al., 2008) and the analysis of the distribution of water ingress for sub-sea XLPE cables has also been researched (Hellesø et al., 2011). Studies about the effects of humidity on the XLPE cable lifetime have been performed in the past. However, existing models are known not to explain the degradation process accurately and the topic is still under active research. Due to this, manufacturers have opted for insulation methods for the cables (Montanari, Mazzanti, and Simoni, 2002; Sanniyati et al., 2016; J.-P. Crine, 1998; Uehara et al., 2014; Qingyue et al., 2019).

When underground High Voltage Alternative Current (HVAC) cables are preserved by outer coverings that prevent moisture intrusion, humidity, oxidation and radiation effects are considered not to be a critical degrading factor anymore (Montanari, Mazzanti, and Simoni, 2002; Sanniyati et al., 2016; Mazzanti, 2009; Bicen, 2017). Currently, the most important degrading factor is considered to be *electrical treeing*, which is caused by the partial discharges that are produced by charges accumulated around cavities in the insulator, generated by the uneven electric fields. This degradation process is known to be both temperature and electric field dependent.

However, the electro-thermal cable degradation models that describe this mechanism are complex and sensitive to multiple environmental and structural factors. Accordingly, these models are often simplified for the purpose of reliability analysis. Some authors have approached the problem by modeling the degradation process based on temperature statistics (Buhari and Kopsidas, 2014). However, the temperature of the cable cannot be considered to be a factor that is independent of the grid conditions. Any switching of the grid conditions can cause variations on the stochastic properties of the expected thermal values, and the rare grid fault events can cause increased cable wear that needs to be considered. Some approaches only consider steady-state thermal conditions or first-order electro-thermal models (Buhari, Levi, and Awadallah, 2016; Mariut et al., 2012), but the use of steady-state analysis for the prognostics of cable lifetime is known to be inaccurate and thermal transition analysis is suggested (Mazzanti, 2009). Other solutions approximate the uncertainties of the grid switching mechanisms that govern the future behavior of the cable with more simple statistical models independent of the cable degradation process. Therefore, these solutions are unable to evaluate accurately long term reliability levels caused by a complex grid behavior (Wallace et al., 2009; Bicen, 2017; Aziz et al., 2014; J. I. Aizpurua, Stewart, et al., 2019; Z. Zhang, Assala, and L. Wu, 2018; Buhari, Levi, and Awadallah, 2016). Additionally, not considering the uncertainties in the degradation models can lead to large lifetime estimation errors, due to the exponential relationship between the thermal behavior and the degradation of cables (Mazzanti, 2007).

In order to perform the prognosis of the cable lifetime, it is necessary to first estimate the current polymer condition. In order to evaluate the instantaneous degradation state of the polymers, condition monitoring procedures are established. Often, these procedures require off-line equipment sampling by putting the equipment out of service to perform tests on the cable or on samples extracted from it (Simmons et al., 2013). Elongation At Break (EAB) tests are one such test that is considered a robust technique for the analysis of the polymer condition (Shumaker et al., 2013).

Multiple non-invasive electrical and acoustic methods have been studied for the on-site monitoring of the cables (Khan et al., 2012; Simmons et al., 2013):

1. Loss angle or $\tan\delta$ measurement
2. Leakage current monitoring for insulation resistance measurement
3. Temperature monitoring
4. Frequency-tuned resonant testing
5. Damped Alternating Current (AC) voltage method
6. Very low frequency voltage testing

7. DC voltage testing
8. Impulse voltage testing
9. Partial Discharge (PD) monitoring
10. Time or Frequency Domain Reflectometry
11. Inductance/Capacitance/Resistance measurements
12. Ultrasonic measurements of sound speed and attenuation

It has been found that electrical measurements may not be sensitive to the degradation of the insulation in practice and impedance based methods require that the cable is disconnected to install instrumentation (Simmons et al., 2013). Loss angle measurements were found to be capable of identifying degradation due to gamma rays, but it has been found that the measurements are not affected by thermal degradation (Shao et al., 2018) and it can be invasive due to the requirement to decouple the cable on the ends. Some methods such as reflectometry can cause increased degradation on the cables (Simmons et al., 2013). Partial discharges are known to be an efficient method for identifying the degradation state of the cables (B. Chen et al., 2017). However, their monitoring and modeling can be challenging because it is a complex and localized effect, and they are mostly employed for incipient fault detection (Alsharif et al., 2009; Jee Keen Raymond, Illias, and Abu Bakar, 2017; Siddiqui et al., 2016). Research has been performed suggesting certain risk level classification from the measurements (Arnold and Kment Attila, Pipa Marek, 2012).

More recently, it has been shown that the degradation state of the cable insulator polymers can be estimated by measuring the temperature applied to the insulator (Mazzanti and Montanari, 1997; Montanari, Mazzanti, and Simoni, 2002; Mazzanti, 2007; Mazzanti, 2009; Marzinotto and Mazzanti, 2014). Laboratory-based cable lifetime predictions based on average temperature and load values have been compared and validated in real installation cable lifetime measurements (Z. Zhang, Assala, and L. Wu, 2018). Therefore, the statistical analysis of the temperature of the polymer can be used to obtain an accurate estimation of the insulator degradation.

During the diagnostics stage the temperature of the insulator can be directly measured if the cables include fiber optics or other temperature sensors (Yilmaz and Karlik, 2006) or indirectly estimated, based on actual load measurements. Methods to obtain an estimation of the thermal rating or ampacity of cables have been known for long (Buller, 1951; Neher, 1964). More recently Dynamic Thermal Rating (DTR) has been employed to offer more accurate load limits depending on the dynamic weather conditions and loads, and to optimize the usage of the grid (Douglass and Edris, 1996; Roberts, Taylor, and Michiorri, 2008; Echavarren, Rouco, and González, 2011; Teh, Lai, and Y.-H. H. Cheng, 2017).

From the available electro-thermal models, the degradation level of the cable can be predicted by estimating the temperature of the cable under given grid and demand conditions, and then estimating the degradation under the estimated temperature values (Mazzanti, 2009). Then, the RUL of a cable can be estimated in the prognostics stage, by applying the probabilistic conditions of the grid to the cable thermal models, in order to evaluate the probable cable temperatures, and the associated degradation trajectories of the cable insulator. The electro-thermal simulations cannot be based on simplified steady-state conditions, and they must perform an accurate simulation of the thermal transients caused by the varying loads applied to the cable, for an accurate degradation estimation (Mazzanti, 2007; Mazzanti, 2009).

This method for the thermal rating and degradation models of cables has been applied to the cable condition monitoring and the prognosis predictions of the cables (Bicen, 2017; Teh, Lai, and Y.-H. H. Cheng, 2017; Buhari, Levi, and Awadallah, 2016). However, due to the complexity of the grid, the models applied for the prognosis are either deterministic, steady-state thermal models, or they are based on simplified statistical models. The prognosis stage requires an accurate probabilistic analysis where the behaviors of the hundreds of components composing the grid are interrelated. Therefore, the events cannot be independently generated for simulation, and accurate transient thermal models must be evaluated.

Accordingly, it is desirable to evaluate the electro-thermal degradation models of the cable under a complete statistical model of the grid, including its switching, power demand and individual component models. The models must also cover the emergency loading conditions, which are known to be one of the main factors for the aging of a grid (Willis, 2004, Chapter 30; Kopsidas and S. Liu, 2018; Mazzanti, 2009).

The uncertainties involving each of these elements need to be considered along with the uncertainty propagation due to the grid structure. Methods for such a propagation of effects are still being evaluated (Coble et al., 2012, Chapter 9; Cadini, Agliardi, and Zio, 2017a).

The stochastic analysis of the grid is often too complex to solve analytically and MC simulations are employed for their numerical solution (G. J. Anders, 2005, Chapter 4). However, failure events in electrical grids are generally rare (because of their high reliability), but once they occur the grids may suffer from cascading effect. Performing an accurate MC analysis of these events demands performing millions of trials, which is impractical for the real time prognostics of equipment (Cadini, Agliardi, and Zio, 2017b; Dueñas-Osorio and Vemuru, 2009).

Certain authors have recently approached the solution for the reliability analysis under this situation by employing multiple sequential independent MC simulations that simplify the analysis (Kopsidas and S. Liu, 2018).

In this context, the objectives for this thesis require:

- A formalism to model failure and operational events of the electric power distribution grids.
- Methods for monitoring the health of the cable under analysis.
- A mechanism for the integration of the component degradation models in a grid model.
- An architecture for performing near real-time analysis of the cable failure probability and prognostics.

The following subsections describe an analysis of the current state of the art in the fields of failure modeling, condition monitoring, degradation models of electric power distribution grid components and architectures for their simulation.

1.3.2 *Degradation models of other components in the grid*

The occurrence rates of the failure and operational events of the grid are affected by the component conditions. These events cause state switching, which vary the rate at which the components degrade.

These varying rates through the component life are often modeled as a Weibull process (Sánchez-Silva and Klutke, 2016b, Chapter 2). However, the method considers events to be independent of the grid behavior. An alternative for a more accurate description of its stochastic behavior is the use of complete accurate models of their internal physics in connection with the grid model.

During the CM stage, actual estimations of the component condition can be obtained from direct or indirect measurements. Research exists for the CM and diagnosis of switchgear such as CBs, transformers and string insulators (D. Zhang et al., 2011; Feizifar and Usta, 2019; Risos and Gouws, 2019; Chakravorti, Dey, and Chatterjee, 2013; Hao et al., 2018; Xiao et al., 2017; Sampedro et al., 2019). For the prognostics stage, either data-driven statistics or physics model-based simulations can be performed, or a combination of both (Kim, An, and Choi, 2017a, Section 1.4). Data-driven degradation models have been widely used both for transformers and CBs (Leone, Cristaldi, and Turrin, 2017; Biswas, Srivastava, and Whitehead, 2015; Trappey et al., 2014; G. Li et al., 2017). Model-based and hybrid degradation models also exist and research on the field is ongoing, which may permit performing longer-term estimations (J. I. Aizpurua, Catterson, Abdulhadi, et al., 2018; Catterson, Melone, and Garcia, 2016; Guo et al., 2018). Little research has been found towards prognostics models for string insulators. However, statistical models have been found for certain faults that may also permit prognostics modeling of these components (Savadjiev and Farzaneh, 2004).

1.3.3 *Grid Model*

The grid model must describe the switching mechanisms of the grid and the electrical and other physical parameters that govern the components or their degradation levels. The definition of these physical condition parameters is necessary for the prognostics analysis of physics-based models.

However, most common reliability analysis formalisms describe failure events with a binary $\{0,1\}$ value. The formalism must be able to represent repairable components and the time relationships between the events, in order to model the cascading event propagation through the grid components.

During the last decades multiple formalisms have been developed for reliability analysis. Most common formalisms used in the reliability analysis literature are Reliability Block Diagram (RBD) and Static Fault Tree (SFT) (Ruijters and Stoelinga, 2015; Kumamoto and Henley, 1996; Abdul Rahman et al., 2013; Volkanovski, Čepin, and Mavko, 2009; Adelabu, Haruna, and Aliyu, 2018). However, SFTs are not capable of representing time relationships between events that define the temporal behavior of a grid. Instead, they only represent the combinational fault event mechanisms that lead to an eventual system failure.

Dynamic Fault Trees (DFTs) and their variants such as Parametric Fault Tree (PFT), Repairable Fault Tree (RFT), Generalized Fault Tree (GFT) and Temporal Fault Tree (TFT) extend SFT to describe time and ordering relationships between the events, that can be employed to represent the cascading effects on power distribution grids, repair or maintenance processes, and for modularization (Čepin and Mavko, 2002; Bobbio et al., 2003; Raiteri et al., 2004; Codetta-Raiteri, 2011; Kabir, Walker, and Papadopoulos, 2014; Palshikar, 2002). In the field of electric power distribution, authors have proposed the use of DFT for reliability analysis of components and the grid interactions (Sihite, 2013; Alves et al., 2019). However, these formalisms consider the events to be Boolean valued, and they can be limiting for the representation of physical models.

Another issue with DFT models is that there is no agreement on the method to represent complex time relationships between events (Palshikar, 2002; Walker and Papadopoulos, 2009; Wijayarathna and Maekawa, 2000; Allen, 1983).

More recently, several formalisms have been developed for the definition of more complex system failure mechanisms. Boolean Driven Markov Process (BDMP) and Generalized Boolean Driven Markov Process (GBDMP) were designed with the familiar Fault Tree (FT) representation in mind that helps an easy understanding of the failure mechanisms. Unlike DFT, the formalisms are defined to be able to represent repairable and redundant equipment and dynamic redundant systems by permitting the system to reconfigure under certain conditions (Bouissou, 2002; Piriou, Faure, and Lesage, 2016). However, the formalisms are unable include complex non-linear functions and processes that describe physics-based degradation models.

Piecewise-Deterministic Markov Process (PDMP) permits the introduction of deterministic behavior of the system by ordinary differential equations and it has been used in tools like PyCATSHOO (Hassane CHRAIBI, 2019) for the analysis of power distribution grids (Chraibi, 2013; Galtier, 2019). However, modeling all the possible states of all the components in a grid can be complex and not optimal for an FPGA transformation.

Adaptive Transition System (ATS) is a formalism presented as a flexible solution to represent more complex failure mechanisms while ensuring a unified representation for common structures (G. A. Manno, 2012; G. Manno et al., 2014). The formalism provides mechanisms for the representation of systems with several interconnected parts. Each individual equipment is represented as *transition systems* which are adaptive thanks to *attributes* that depend on the state of other transition systems.

Stochastic Petri Nets (SPNs) have been frequently used for the analysis of power distributions and their reliability analysis (Calderaro et al., 2011; Volovoi et al., n.d.). Certain variants of Petri nets, such as Generally Distributed Transition Stochastic Petri Nets (GDT_SPNs) and Fluid Stochastic Petri Nets (FSPNs) extend the formalism for non-exponential arbitrary distributions and fixed time intervals, and to the representation of fluid flow (Rogge-Solti, Aalst, and Weske, 2013; Horton et al., 1998).

Stochastic Activity Network (SAN) is another Petri Net variant that predates ATS, originally developed to perform performance analyses of very complex systems (Sanders and Meyer, 2001). Some authors found that SAN was too flexible developed a more rigid formalism ATS to force unifying structure on the reliability models (G. A. Manno, 2012). However, SAN has been demonstrated to be a very flexible formalism for reliability analysis and prognosis of systems that include degrading equipment and complex inter-process dependencies (Sanders and Meyer, 2001; Maza, 2015; J. I. Aizpurua and Catterson, 2016). SAN applies very few restrictions on the model and it is capable of operating with very complex functions necessary to describe certain physics-based models. This formalism has been, therefore, adopted as the base for this research, and a more complete description of the formalism has been provided in Section 2.2 for a better comprehension.

1.3.4 *FPGA and other architectures for reliability analysis*

For larger and more complex grid models that include components with deterministic and arbitrarily distributed stochastic events and mechanisms that inter-relate the components, generally no analytical solution exists for the transient probability values of an event or a system state. In those situations, numerical MC simulations are employed to analyze the system failure probability (Kabir, 2017; Zio, 2016). However, grid models can be large, with components that evolve independently but inter-connected by time-related mechanisms.

The operational events that govern the components are generally short-time events (e.g. time that takes a switch to maneuver). However, the stochastic intervals between these events are frequently long, especially on high-reliability equipment where fault probabilities are very low. An accurate representation of these rare events requires long simulation times and large computing resources, due to the large number of MC trials to represent these uncertainties and the long mission time of each trial (Pan, Z. Wang, and Sun, 2015).

In order to accelerate the simulations, Graphics Processing Units (GPUs) have recently been employed for Petri net simulation, but GPUs do not perform optimally for the parallelization of irregular structures and they require adapting the Petri Net (PN) and careful planning for parallel sub-net simulation and reduced memory resources (Nurvitadhi et al., 2017; Yianni et al., 2018). The method cannot generally be used for SAN models, due to dependencies of transitions (*activities* in SAN) behavior with the marking of other subprocesses (see Section 2.2). Large GPUs with thousands of cores exist, but these GPUs and the required memory resources are not generally available on low-cost on-site embedded devices.

Unlike GPUs, FPGA implementations are inherently distributed and connected in hardware. Accordingly, a direct FPGA translation of a network structure without optimization is possible if its governing algorithms can be integrated.

Petri nets have been successfully synthesized in FPGA logic for purposes such as simulation or control (Leroux, Andreu, and Godary-Dejean, 2015; Bukowiec and Adamski, 2012). Markov Chain Monte Carlo (MCMC) have also been implemented and accelerated on FPGA (S. Liu, Mingas, and Bouganis, 2017). However, to the best of our knowledge, the use of FPGA for MC reliability analysis has been limited to non-state-space FT models, DFT, or hybrid fuzzy FT (Aliee and Zarandi, 2013; Shoar et al., 2017; Ejlali and Miremadi, 2004; Zhu et al., 2017). However, they cannot describe complex inter-process dependencies or time-varying models with degradation mechanisms that require employing more complex state-space models such as Petri nets and their variants, such as SAN (Ruijters and Stoelinga, 2015; Piriou, Faure, and Lesage, 2016; Song and Schnieder, 2018; J. I. Aizpurua and Catterson, 2016).

1.4 DECLARATION OF EARLIER PUBLICATIONS

The following publications have been produced from the results obtained from through this thesis, and some figures, results and materials have been used for the writing of this document:

Garro, U. et al. (Apr. 2019a). “FPGA-Based Degradation and Reliability Monitor for Underground Cables”. In: *Sensors* 19.9, p. 1995. ISSN: 1424-8220. DOI: `10.3390/s19091995`. URL: `https://www.mdpi.com/1424-8220/19/9/1995`

Garro, U. et al. (July 2019b). “FPGA-Based Stochastic Activity Networks for On-Line Reliability Monitoring”. In: *IEEE Transactions on Industrial Electronics*, pp. 1–1. ISSN: 0278-0046. DOI: `10.1109/TIE.2019.2928244`. URL: `https://ieeexplore.ieee.org/document/8765588/`

1.5 ORGANIZATION OF THE THESIS

Figure 1.3 summarizes the organization and results obtained from this thesis. The following are the contents of each chapter:

- Chapter 2: *Grid Failure Modeling and Prognostics Analysis*

  This chapter describes a methodology for the analysis and evaluation of the reliability of a grid component, based on a SAN model of the grid, capable of including physics-based component degradation models.

- Chapter 3: *Condition Monitoring and Prognostics of Underground Cables and Other Grid Equipment*

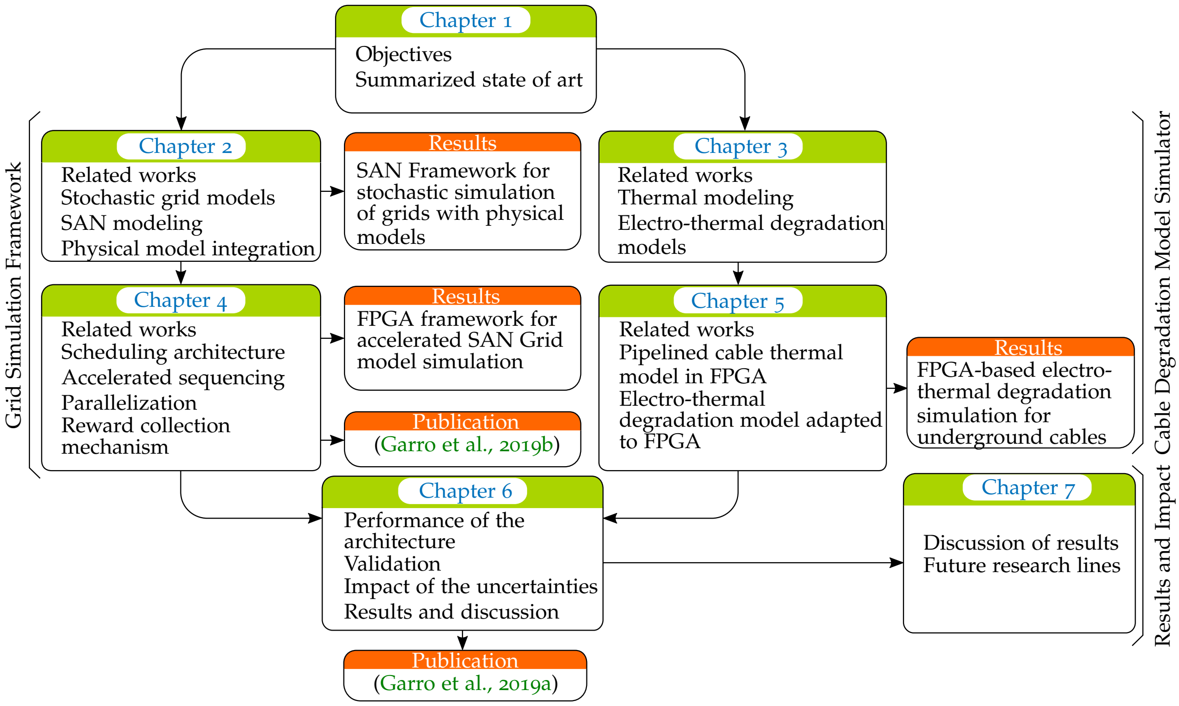


Figure 1.3: Organization of the document and results.

This chapter describes the electro-thermal heating and degradation models and methods adopted for this thesis.

- Chapter 4: *MC Simulation of SAN With Physics-Based Model Activites, Based on FPGA*

  This chapter describes an accelerated FPGA framework for the simulation of SAN models, capable of including physics-based models.

- Chapter 5: *FPGA-Based Accelerated Electro-Thermal Degradation Model of Cables*

  This chapter describes a pipelined FPGA architecture for the accelerated simulation of the cable degradation model physics described in Chapter 3.

- Chapter 6: *Diagnostics and Prognosis of the Cable based on FPGA*

  This chapter validates the developed architecture for the condition monitoring and prognosis of underground cables and analyzes the impact of the method on the cable lifetime predictions.

- And finally, Chapter 7: *Conclusions and Suggested Future Research Lines*

  This chapter discusses the results and conclusions that can be obtained from the present thesis and proposes future research lines in this research field.

2

GRID FAILURE MODELING AND PROGNOSTICS ANALYSIS

In the context of prognosis analysis of the equipment, the future evolution of equipment is estimated from past measurements and statistically generated future system inputs (Zio, 2016). Prognostics approaches generally fall in two groups, or a mixture of the two (Kim, An, and Choi, 2017a, Section 1.4):

1. Data-driven models.
2. Physics-model based prediction of the future behavior.

The former group employs past data, to extrapolate the future distributions and to predict the degradation (or recovery) trend of the system (M. Orchard, Tobar, and G. Vachtsevanos, 2009; Engel et al., 2000).

The physics-based modeling method requires an accurate knowledge of the degradation model that governs the system and its associated parameters, which can be complex (e.g. modeling of the heating of an underground cable, or the humidity evaporation of the soil where the cable is laid). The models have to describe the mechanisms that degrade the system, and apply the expected inputs to the system in order to estimate the time at which the failure event will occur.

An advantage of physics-based models is that they can not only be used to monitor the current condition of the degrading equipment by using measured data as input, but also for predicting the future trends of degradation based on probabilistic inputs.

However, the flexibility of common reliability analysis formalisms to introduce physics-based models can be limited.

2.1 A FORMALISM FOR CABLE DEGRADATION AND FAILURE MODELING

A cable degradation and failure model should be capable of representing all the events and parameters that affect its degradation process. Therefore, the reliability analysis of the cable should be capable to include:

1. The physics-based model that describes the electro-thermal degradation mechanism of the cable (see Chapter 4 for a detailed description);
2. The failure mechanisms that describe the system failure condition (e.g. multiple redundant cables);
3. The power grid mechanisms that describe the inputs that drive the physical cable model (e.g. switches, transformers, grid load etc.);
4. The mechanisms that describe the relationship of the cable behavior with the grid (e.g. maintenance procedures, reduction of maximum load due to the cable degradation, behavior when the cable is failed, etc.).

The representation of all those features requires representing very complex time varying processes, that often include non-linear functions.

As it was discussed in Section 1.3.3. Stochastic Activity Network (SAN) is a formalism that permits the simulation of models with very complex functions describing their behavior and time intervals that follow non-standard probability distributions.

More recently, the formalism has been shown to be very flexible for grid modeling of cascading failures (Chiaradonna, Lollini, and Di Giandomenico, 2007) and for reliability and prognosis analysis (J. Aizpurua and Muxika, 2015; Maza, 2015; J. I. Aizpurua, Catterson, Papadopoulos, et al., 2017). Therefore, the SAN formalism, described in the following sections, has been adopted as the base for the research models for this thesis. However, obtaining the analytical solution of a SAN model is frequently not possible, and Monte Carlo (MC) simulations are employed for obtaining the metrics (Sanders, 1988).

2.2 THE SAN FORMALISM

SAN are a class of Stochastic Petri Net (SPN) which was introduced in (Movaghar, 1984; Sanders and Meyer, 1991a; Sanders and Meyer, 2001), originally designed for performance analysis. (Movaghar, 2001) provided a formal definition that allows the formalism to be subjected to analysis.

In SPNs and in most its derivative formalisms such as Generalized Stochastic Petri Net (GSPN), the different possible conditions of the system or individual components are represented by *places*. These places have a *marking* associated to them, generally representing that a certain condition or event has been activated or how many times it has been processed (although the meaning of the marking is flexible and not defined by the formalism itself).

In SAN, the state of the model is represented with places P that hold a marking μ_{SAN} that are similar to *places* in SPN.

The original SAN formalism adopted the same concept of marking of SPN (Movaghar, 2001). Later, (Azgomi and Movaghar, 2003b) developed the concept of Coloured Stochastic Activity Network (CSAN) that extended the formalism by describing markings of any type, including non-integer or vector types. The term *extended place* is used in software simulation tools such as Möbius Tool for places with non-integer marking (*The Möbius Tool* 2017).

The switching between the different conditions or states (different markings) is governed by a set of activities A. Activities are generally used to represent possible events that cause the system to switch between different states. When activities are triggered, the markings of the associated places are altered by performing operations on the place markings. The operations are defined by *functions* that can describe complex behavior such as degradation mechanism physics. Functions are not defined by the activities themselves, but in *gates* that are associated to the activity.

The gates not only change the markings of the places, but they also define when an activity can be triggered (that is, in which system state an event is possible). Two types of gates are defined: input gates (ig) or output gates (og). While both input and output gates define functions that alter the system marking, only input gates in a SAN model control when an activity is *enabled* or disabled. That is, they define when the associated event can be triggered or not. Figure 2.1 shows the graphical symbols used for each element in a SAN model.

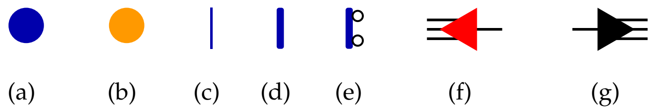


Figure 2.1: Symbols used in SAN formalism as defined in (Sanders and Meyer, 1991a): (a) Place; (b) Extended Place; (c) Instantaneous activity; (d) Timed Activity (with single case); (e) Timed Activity (with two cases); (f) Input Gate and (g) Output Gate.

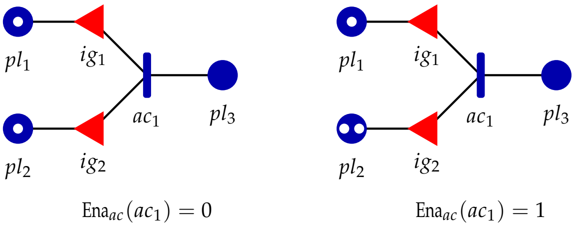
In a SAN model, each activity ac_i is associated to a set of output gates $\mathrm{OG}(ac_i)$, and a set of input gates $\mathrm{IG}(ac_i)$. When an activity is *completed* (triggered), the functions defined by both the associated input and output gates are executed, altering the marking of the places associated to these gates. Each input gate ig_i can have a $Pl(ig_i)$ set of connected (associated) input places, and each output gate og_j can have a $Pl(og_j)$ set of output places.

In the original SAN formalism, the completion of an activity could have multiple possible outcomes called *cases* with an associated probability distribution, which is represented as shown in Figure 2.1e. In the formalism redefinition in (Movaghar, 2001), multiple cases are replaced by instantaneous activities instead, and therefore multiple case activities have not been considered through this thesis implementations.

In addition to the associated function, each input gate ig_i also has a Boolean *predicate* function associated. This predicate function, denoted $\mathrm{Pred}_{ig}(ig_i)$ defines if an activity can be *enabled* for the current state. An activity ac_i is said to become *active* or *enabled* ($\mathrm{Ena}_{ac}(ac_i) = 1$) when the predicate functions of all the associated input gates are met. That is:

$$\mathrm{Ena}_{ac}(ac_i) = \begin{cases} 1, & \mathrm{Pred}_{ig}(ig_j) = 1 \;,\; \forall ig_j \in \mathrm{IG}(ac_i) \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

Figure 2.2 shows an example of activity enablement predicate functions, where $\mathrm{m}(pl_k)$ denotes the marking of place pl_k.

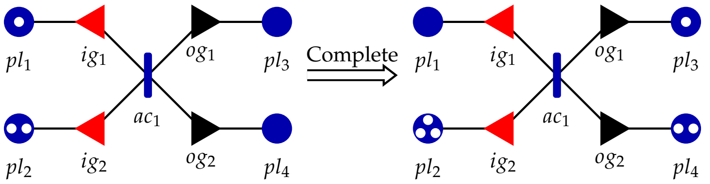


Gate	$\mathrm{Pred}_{ig}()$
ig_1	$\{\mathrm{m}(pl_1) \geq 1\}$
ig_2	$\{\mathrm{m}(pl_2) \geq 2\}$

Figure 2.2: Activity enablement example.

In this example, activity ac_1 is enabled when the marking of place pl_1 is greater than or equal to 1 and the marking of pl_2 is greater than or equal to 2 at the same time.

Once an activity ac_i is enabled, a completion time interval $\mathrm{ct}(ac_i)$ is generated for it. At that time, unless it is disabled earlier, the activity *completes* by triggering first the functions of the input gates, and then the functions of the output gates, as shown in Figure 2.3[1]. $f_g(ig_i)$ and $f_g(og_j)$ denote the associated functions of input gate ig_i and og_j respectively.



Gate	$\mathrm{Pred}_{ig}()$
ig_1	$\mathrm{m}(pl_1) \geq 1$
ig_2	$\mathrm{m}(pl_2) \geq 2$

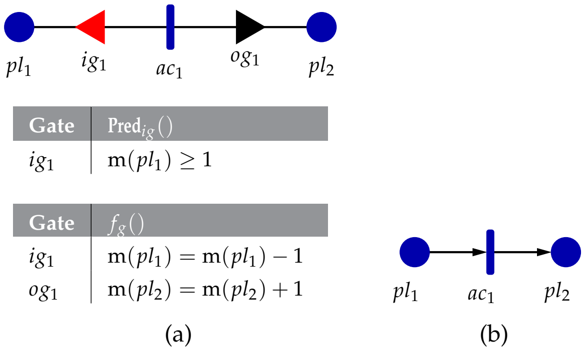
Gate	$f_g()$
ig_1	$\mathrm{m}(pl_1) = \mathrm{m}(pl_1) - 1$
ig_2	$\mathrm{m}(pl_2) = \mathrm{m}(pl_2) + 1$
og_1	$\mathrm{m}(pl_3) = \mathrm{m}(pl_3) + 1$
og_1	$\mathrm{m}(pl_4) = 2$

Figure 2.3: Activity completion example.

---

[1] In SAN no ordering is specified for applying the functions associated with the multiple input and output gates, other than applying input functions before output functions (Sanders and Meyer, 2001).

When the input and output gate function make the activity completion to behave similarly to SPN, that is, input gate functions decrement the number of marks in the associated places by 1, and output gates increment the marking of the associated places by 1, the gates are not drawn for clarity. In this case, gates are represented as directed lines as shown in Figure 2.4.



Gate	$\text{Pred}_{ig}()$
ig_1	$\text{m}(pl_1) \geq 1$

Gate	$f_g()$
ig_1	$\text{m}(pl_1) = \text{m}(pl_1) - 1$
og_1	$\text{m}(pl_2) = \text{m}(pl_2) + 1$

(a) (b)

Figure 2.4: (a) SAN model with input and output gates, desribing a SPN-like behavior and (b) the equivalent simplified SAN representation with directed lines.

Each activity is associated to a *distribution function* $f_{ac}(ac_i)$ and a *reactivation function* $\text{React}_{ac}(ac_i)$. Once the activity becomes active, a new time interval τ_{ac_i} is generated according to the distribution function, and a new interval τ'_{ac_i} is generated overriding the previous generated interval, whenever the reactivation function is met. Therefore, after the activation of an activity, this activity can be:

1. *Completed* after this time interval;
2. *Aborted* by the completion of another activity that alters the marking, resulting with one or more of the input gates not meeting the predicate condition;
3. *Reactivated* when the reactivation function is met, causing a new time interval to be calculated for the activity.

Figure 2.5 shows the different possible behaviors of an activity after its enablement (Sanders and Meyer, 2001)

Each activity can have one or more *cases* associated that represent the uncertainty of different possible outcomes after an activity is completed. Each case, represented by circles in the activity as shown in Figure 2.1e, completes by executing the functions of a different output gate set that are associated with the case.

Therefore, an *Activity Network* is defined as (Movaghar, 1984):

$$AN = (P, A, I, O, c, \nu, \iota, o) \tag{2.2}$$

where P are the set of places of the network, A the set of activities, I is the set of input gates, O the set of output gates, ι and o map input and output gate connections respectively, ν is the set of activity types (which are either *timed* or *instantaneous*) and c is set of the *cases* of an activity.

The formalism definition described in (Movaghar, 2001) replaces *cases* by a set of instant activities and the formalism is simplified as in Equation (2.3) that has been adopted for this thesis

$$AN = (P, A, I, O, \nu, \iota, o) \tag{2.3}$$

SAN models can be structured by connecting multiple SAN networks by means of additional construction elements. When the networks only contain activities, places, input

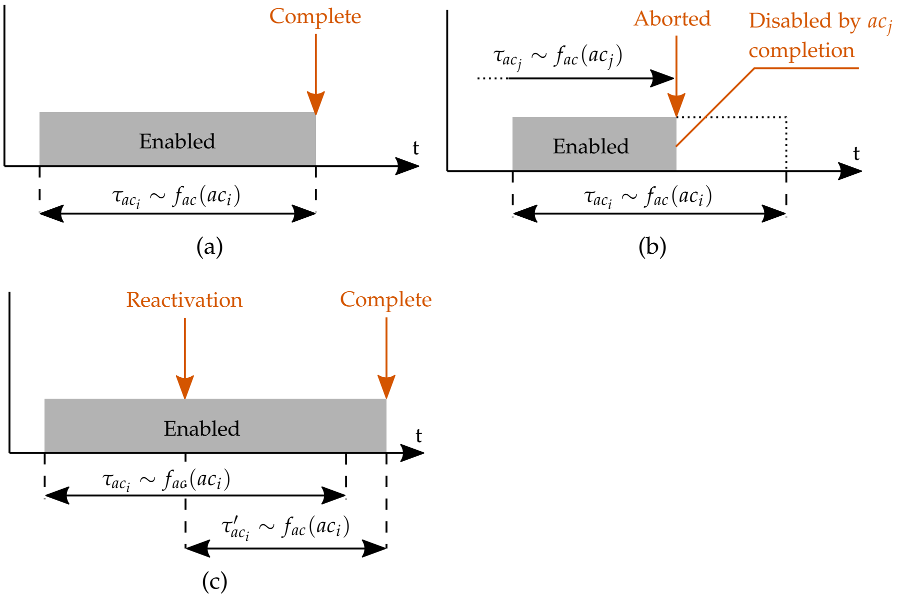


Figure 2.5: Possible behavior of an activity after its enablement: (a) completion of the activity after the evaluated time interval, (a) the ac_i completion is aborted by the completion of activity ac_j and (c) the completion time of the activity is regenerated when the condition of the reactivation function is met.

gates and output gates, they are called called *atomic* models. Different methods have been defined for creating structured SAN models:

1. (Sanders and Meyer, 1991a) described the use of *JOIN* and *REP* (replicate) procedures. The former permits connecting places in multiple atomic SAN models, that behave as a single place. *REP* permits replicating identical atomic SAN submodels that share certain places, in order to represent replicate or redundant equipment. The extension permits reducing the size of the models for larger complex and redundant systems.

2. (Azgomi, Movaghar, et al., 2005) described the construction of hierarchical SAN, that can be employed to construct more complex, non-tree shaped structured SAN models.

The Hierarchical Stochastic Activity Network (HSAN) formalism can be used to represent models that cannot be structured employing the *REP* and *JOIN* procedures, e.g. ring topologies. HSAN permits a convenient top-down definition for component abstraction and reuse. The formalism adds the definition of Macro Activity Class (MAC), which are interconnectable SAN networks where each MAC behaves as a complex activity on its own. The MACs are interconnected between them on their input and output *fusion place*s. Their definition is given as:

$$MAC = (SAN, IFP, OFP) \tag{2.4}$$

where SAN is an activity network model that can contain places, gates, activities and other macro activities. IFP is the set the input fusion places and OFP is the set of output fusion places. Their graphical representation is shown in Figure 2.6.

The HSAN activity network is described as a set of SAN, MAC and a *fusion function FF* as:

$$HSAN = (SAN, MACS, MA, FF) \tag{2.5}$$

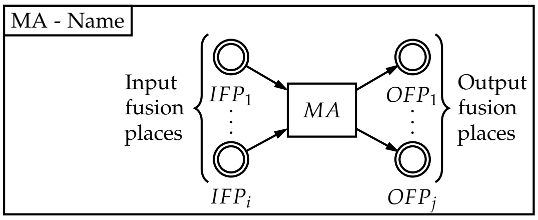


Figure 2.6: Macro Activity Class (MAC) definition

where SAN is a SAN model defined as in Equation (2.3), $MACS$ is a set of MACs as defined in Equation (2.4) , MA is a set of *macro activities* associated with one of the MACs and FF is a fusion function as $FF : MA \times FP \rightarrow P$ that fuse fusion places the set of fusion places FP to the set of places into a place set P in the compound SAN model, resulting a hierarchical composition as shown in Figure 2.7

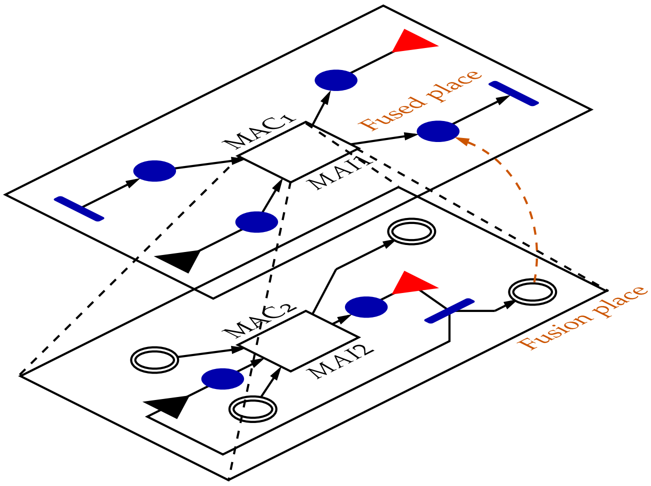


Figure 2.7: Fusion of places in a HSAN model, including instances $MAI1$ and $MAI2$ of MACs $MAC1$ and $MAC2$.

The HSAN formalism also eliminates certain restrictions of original SAN such as the requirement to check the *well-behavedness* which is complex to evaluate (Sanders and Meyer, 2001). The requirement for an HSAN to be solvable either analytically or via simulation is that it is *well-defined*, that is, there are no nested cyclic loops and it is finite. Any well-defined HSAN can be converted into a flat SAN network as described in Equation (2.3) (Azgomi and Movaghar, 2003a).

It is important to notice that in a HSAN model, despite the hierarchical structure, a single atomic activity will be completed at each given instant, and if two activities are enabled at the same time, the choice of which activity will complete first will be made stochastically.

Therefore, in terms of performance and capabilities of the different SAN formalisms their simulation is similar. Unless the simulations take advantage of independent parts in the replicates of a model constructed with *JOIN/REP* model structures, performing the simulations in a HSAN model or a flat atomic model would result in the same accuracy and simulation times, given that a single activity can be triggered at each instant, and the simulation is to be executed sequentially by the definition of the formalism (Azgomi, Movaghar, et al., 2005). Accordingly, a hierarchical SAN model simulation approach has been considered as the base for the present thesis, due to their flexibility.

Reward functions are employed to analyze SAN models by measuring the rates or statuses of a given place (Sanders and Meyer, 1991b). Reward functions are defined by relating a *reward rate* to each state, which accumulates on the variable while on that state. The feature can be used e.g. for the purpose of performance, degradation, cost of maintenance, component replacement or availability analysis.

A software solution by the University of Illinois called *Möbius Tool* exists for the simulation of the SAN models (*The Möbius Tool* 2017). However, while the tool supports the simulation of composed models employing *JOIN* and *REP* procedures and extended places for non-integer type markings, at the time of the writing of thesis document, no tool exists that supports the simulation of HSAN. However, as any *well-defined* SAN model can be converted into a flat SAN model, any SAN simulator can be employed to obtain their solution.

2.3 DYNAMICS OF PHYSICS-BASED MODELS WITHIN A SAN MODEL

The integration of a physics-based system or component model within a SAN model requires that:

1. The physics-based model must be able to differentiate the different conditions or states (*marking* in a SAN model) and simulate the physics according to those conditions
2. The state of the physics-based models must be able to alter the behavior of the system, e.g. the grid condition or the maintenance process.

Accordingly, the physics-based models can be integrated into a SAN model either as:

1. Functions of the input gates and/or output gates that depend on the current marking and alter an output marking (e.g. cable temperature or failure state markings).
2. A timed activity that alters its completion time, according to the physics-based model simulation, causing changes in the SAN model.

In any of the two cases, given that the future behavior of a physics-based model generally depends on the past system behavior, the physics-based model must store the current state statically throughout the simulation. Therefore, the physics-based models cannot be implemented as memory-less functions.

2.3.1 *Physics-based models as a gate*

If the physics-based model is integrated as a gate (either input or output gate) as shown in Figure 2.8, the model function will only be evaluated when the corresponding associated activity ac_{τ_s} completes. In this structure, activity ac_{τ_s} governs the execution of the output

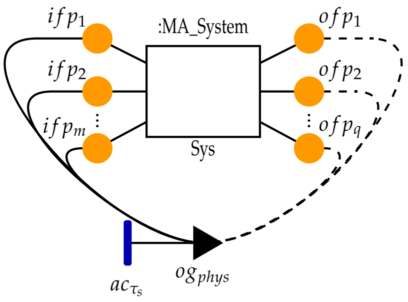


Figure 2.8: Physics-based model integrated within a SAN model as an output gate.

gate og_{phys} corresponding to the physics-based model. The activity will typically complete at a fixed rate of once per sampling interval τ_s.

The output gate then evaluates the associated function that describes the physics-based model. The outputs of this physics model, e.g. the degradation state of a transformer, are the input fusion places ifp_i of the system macro-activity instance. This MAC can e.g. represent the grid structure.

Therefore, the physics-based model alters system model MAC behavior through the input fusion places denoted ifp_i. Additionally, the physics-based models generally depend on the system condition, e.g. the load of a transformer is dependent on the current grid condition. Therefore, the output gate functions depend on the instantaneous system state. In this structure, the physics-based model is dependent on a set of output fusion-places denoted ofp_j. The dashed lines indicate that the gate does not alter the output fusion places, but it depends on their current condition.

The activity ac_{τ_s} produces deterministic time intervals with the value of the sampling time interval required by the physics-based model.

In this structure the activity needs to be triggered every sampling interval τ_s, and therefore the output fusion places (inputs to the physics-based model) need to be re-evaluated every sampling interval in the simulation, and the corresponding activity completed.

2.3.2 *Physics-based models as an activity*

When the physics-based model is integrated as a SAN activity, a stochastically and/or deterministically evaluated time interval (e.g. a degradation or a failure interval) is calculated based on the system state, which is obtained from the output fusion places $ofp_1 \dots ofp_q$, as shown in Figure 2.9. In this structure, the activity itself decides when it will need to be re-evaluated (e.g. when the system state is changed) and a new time interval is produced, that defines when the next re-evaluation will occur.

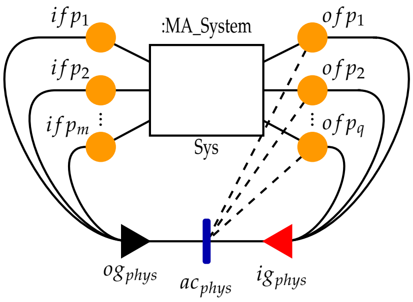


Figure 2.9: Physics-based model integrated within a SAN model as an activity.

However, in this architecture, the activity completion competes with other activities in the grid, which may *abort* the completion time planned by physics-based model activity ac_{phys} as shown in Figure 2.10.

At this point, the simulation performed between the time instant when the activity is aborted t_{abort} and $t_{complete}$ needs to be reverted and the activity completion time re-evaluated, given that the system state could have changed and the simulation thus becomes invalid.

This re-evaluation can be modeled by means of several mechanisms:

1. A reactivation function in ac_{phys} that forces the re-evaluation at any system change that affects the activity.

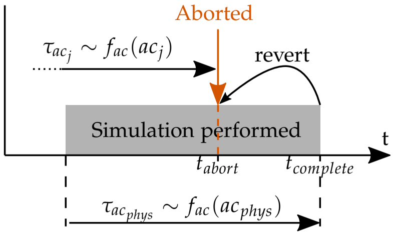


Figure 2.10: The completion of the physical activity can be aborted by other activities that alter the system state.

2. Limiting the simulation to the time before any abort sequencing can occur due to state changes.

The reactivation method described in Section 2.2 requires the continuous monitoring of the system marking for any change, and reverting the system state to the state before the abort time t_{abort}. Given that in many physics-based models there is no analytical means to revert the model and obtaining the previous state, it would be necessary to record every sample of the system state. Therefore, this method is impractical.

The second method requires that the ac_{phys} completion time is evaluated only after the rest of activities are evaluated, and the simulation time to be limited by the minimum of the set of completion times. That is:

$$\tau_{ac_{phys}} = \min \{t_{M,\text{Phys}}, t_{complete}\} \tag{2.6}$$

$$\tau_{complete} \sim f_{ac}(ac_{phys}, t_{M,\text{Phys}}) \tag{2.7}$$

$$t_{M,\text{Phys}} = \min \{t_{ac_i}\}, ac_i \in A \setminus ac_{phys} \tag{2.8}$$

where $t_{M,\text{Phys}}$ is the mission time given to the physics-based activity simulation, which is limited by the rest of activity completion times, $t_{complete}$ is the absolute time referenced completion time (current simulation time plus the completion time) that would result for the physics-based activity to change its state for the given mission time, t_{ac_i} corresponds to the absolute time reference completion times of each activity ac_i excluding ac_{phys}, and $f_{ac}(ac_{phys}, t_{M,\text{Phys}})$ denotes the evaluation of the completion time by the physics-based system model, limited by the partial mission time $t_{M,\text{Phys}}$.

2.4 Modeling of the Grid and the Degrading Equipment in SAN

When applying the method described in Section 2.3 for the integration of the physics-based models of grid equipment, it should be noted that most of the components will be degrading in time, and not all components have analytical models for their physics-based degradation mechanisms.

Often, the failure time of the components is considered Weibull distributed (Lindquist, Bertling, and Eriksson, 2008) or obtained by data-driven techniques that result in a given Probability Density Function (PDF) describing its behavior (J. I. Aizpurua, Catterson, Papadopoulos, et al., 2017).

When a component failure probability distribution is estimated employing data-driven techniques that result in a given PDF, the component can be integrated as a single activity with the given PDF for time interval. However, when its analytical solution is not viable but its physics-based model is known, it can be integrated into the model as described in Section 2.3, resulting a model as shown in Figure 2.11.

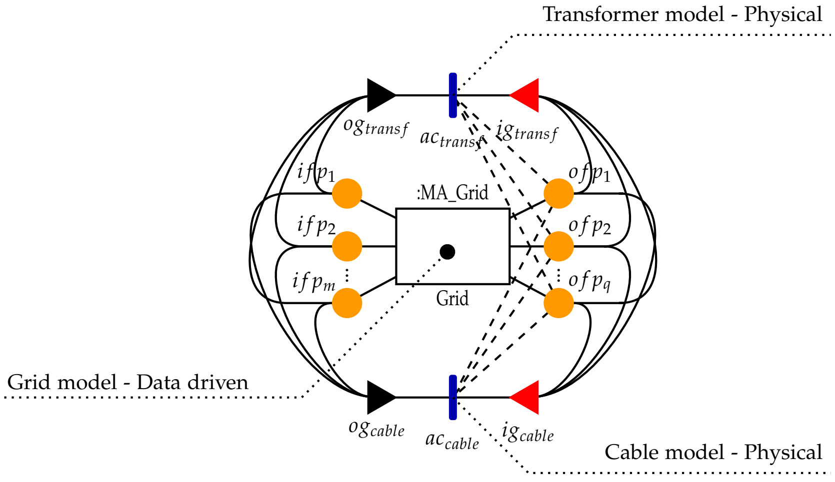


Figure 2.11: Integration of two physical component degradation models within a SAN model of the grid.

It is important to notice that the physics-based models often will not depend on all the output fusion places of the grid, nor alter all the input fusion places of the grid. Each of the components will depend on and alter a given set of the place markings from the grid. However, all component activities may be defined to alter the same markings. The SAN formalism permits that a given marking is altered by multiple activities. However, there is no risk of multiple component models simultaneously altering the same marking, given that the formalism defines that a single activity is completed at each time instant (Sanders and Meyer, 2001).

Figure 2.12 shows an example of a simple grid structure with a transformer tf_1, a circuit breaker cb_1 and a cable cbl_1. It is assumed that physics-based models for the degradation process of the cable and the transformer are known, and the circuit breaker switches according to a statistical model designed from previously collected monitoring data, degrading at each switching event until it completely fails and stops switching.

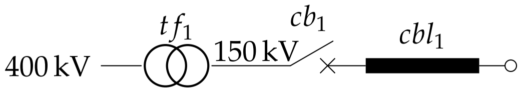
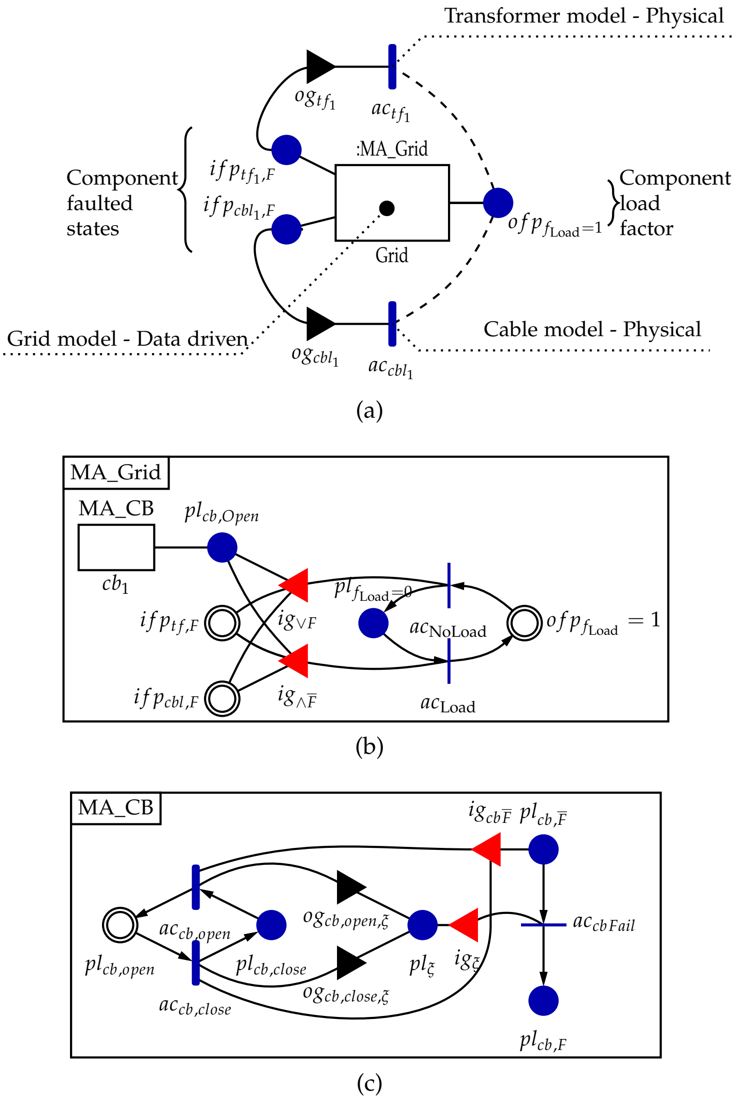


Figure 2.12: Electric power distribution grid use case.

The structure is modeled as shown in Figure 2.13a, where ac_{cbl_1} represents the physics-based model for the Time To Failure (TTF) of the cable and ac_{tf_1} is the physics-based model for the TTF of the transformer. *Grid* is an instance of the *MA_Grid* MAC that models the grid, and provides the instantaneous load factor f_{Load} applied to both the transformer and the cable in $ofp_{f_{\text{Load}}} = 1$, and receives as input the faulted F (or not faulted $\overline{F}$) conditions from the physics-based component models in fused places $ifp_{tf_1,F}$ and $ifp_{cbl_1,F}$.

Figure 2.13b shows the structure of the grid MAC, that models the load that each component will be applied to each component depending on the condition of each component. A normalized load factor of 1 is applied $\text{m}(ofp_{f_{\text{Load}}}) = 1$ if the circuit breaker is closed and both the cable and the transformer are in working condition ($\text{Pred}_{ig}(ig_{\wedge \overline{F}}])$), and a load factor of 0 if the circuit breaker is open or either the cable or the transformer reached their failure state ($\text{Pred}_{ig}(ig_{\vee F})$).

Figure 2.13c models the statistical switching of the circuit breaker with activities $ac_{cb,open}$ and $ac_{cb,close}$. Each time the Circuit Breaker (CB) switches, the degradation level is accumulated in pl_{ξ} by output gates $og_{cb,open,\xi}$ $og_{cb,close,\xi}$. Once the CB reaches its degradation limit



(a)

(b)

(c)

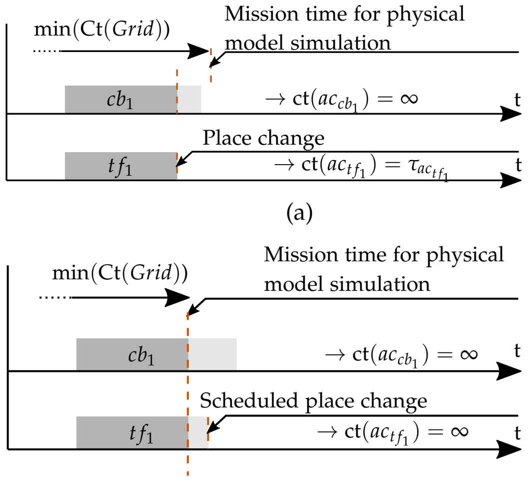
Figure 2.13: HSAN model of the grid structure in Figure 2.12. (a) complete model integrating physics-based component models and the grid model, (b) Model of the grid including the circuit breaker MAC instance and (c) circuit breaker MAC.

$(\text{Pred}_{ig}(ig_{\xi}) = 1)$, the component enters the faulted state $pl_{cb,F}$, that prevents the CB from switching $(\text{Pred}_{ig}(ig_{cb\overline{F}}) = 0)$.

Each time the condition of the grid changes, the behavior of the components with physical models can be altered. Therefore,

1. The simulation of the physical components must be stopped at each grid HSAN state change.
2. All activities in the grid model must be evaluated for their completion time.
3. The minimum completion time of the activities in the grid MAC is calculated.
4. The physically modeled components must be simulated synchronized in time, until one of the physical components causes a state change in one of the fused places, or the minimum completion time of the activities in the grid MAC is reached.
5. If a physically modeled component causes a state change, the completion time for the component causing the state change is given, while the other physically modeled components produce a completion time at infinite in their activities.

The physically modeled activities have no input gates connected to them. Therefore, they are always enabled because they alter their own completion time (usually the failure time) depending on the instantaneous condition of the system described by its marking μ_{SAN}. Therefore, they can reevaluate their completion times after the completion of any other activity, that includes the completion times of the activities in the grid MAC $\mathrm{Ct}(Grid)$, or the completion times of other physical activities. Figure 2.14 shows an example with two activities representing physics-based models of a circuit breaker cb_1 and a transformer tf_1, each with their completion times $\mathrm{ct}(cb_1)$ and $\mathrm{ct}(tf_1)$. When an activity other than the one being simulated completes before the simulation changes the output place markings, the simulation cannot evaluate its completion time, and it will temporarily behave by reporting the completion time as infinity (∞), and it will be reevaluated after the new state change.



(a)

(b)

Figure 2.14: Evaluation of the completion time of the activities of the physically modeled components: (a) when a component completes (changes states) before the mission time defined by the grid activities and (b) when one of the activities in the grid activities completes before any of the physical components change the output state.

2.5 ANALYTICAL SOLUTIONS

Each SAN model place can have multiple possible states depending on its marking. Therefore, even simple SAN models can represent infinite state-space models such as Continuous Time Markov Chain (CTMC) where all the activities are represented by exponential distributions or Generalized Semi-Markov Process (GSMP), where the intervals between each state jumps can be arbitrarily distributed, and each state has multiple possible events (activities) that compete to occur (complete) (Whitt, 1980; Movaghar, 2001).

The objective of the reliability analysis is to obtain the failure probability of the system. It is important to notice that the failure probability varies in time. For example, it is very unlikely that a component in the grid will fail due to the degradation right after it is installed, but it may be more likely that it will break down in a very short time if the component is almost completely degraded after several years of use. Those systems do not reach a stationary failure probability state, unless maintenance processes are put in place. And even then, the failure probability is not constant in time. Therefore, it is important to notice that a transient failure probability value is required as solution, rather than a stationary state solution for the systems.

When all the timed activities in the model are exponentially distributed, the system can be represented and solved as a CTMC (Movaghar, 2001; Cerdà-Alabern, 2014). However, obtaining analytical solutions to more complex models is generally not possible, and numerical solutions are employed. Even when only exponential distributions are involved, when the models are large as in the analysis of a full electric power distribution system model, analytical methods suffer from an state explosion when translating into Markov model states, due to the large number of combinations of states that each component can be into, which increases exponentially with the number of components (2^N in the case of N electric power distribution network components, if each of the components can only be in two states: failed or not failed, and even more complex if more states are represented for each component such as under maintenance, warm standby or cold standby states).

Physics-based models permit the analysis of components whose failure PDF is unknown. Generally, the future state defined by these processes is dependent on the past states. For example, a convolution process considers all past inputs to the system, or in the case of degradation processes the new degradation state will depend not only on the events causing the degradation but also the accumulated degradation. Modeling and producing analytical solutions for those models is therefore considered to be too complex and numerical solutions are required.

2.6 Monte Carlo Simulations and Confidence Margins

MC simulations are a common numerical method to obtain the probability distributions of complex discrete-event systems (Doucet, Freitas, and Gordon, 2001). This simulation method has long been used in reliability analysis (Goldfeld and Dubi, 1987; Y.-F. Wu and Lewins, 1992). While limited in the past to areas such as nuclear power due to the computational requirements, the method is very widely used nowadays in multiple research areas, including reliability and risk analysis (Zio, 2013) and electric power distribution system analysis (Billinton and W. Li, 1994).

The MC method makes no assumptions about the probability distribution that governs the system. It is based on performing a large number N_{MC} of *trials* which are random walks through the possible $\{\text{state}, \text{time}\}$ combinations of the system (Zio, 2013). Each trial can cover long simulation times for each history, until the *mission time* or certain given condition is reached in each simulation. Samples of these simulations are taken through the simulation, and a probabilistic analysis is performed on the results, in order to obtain the PDF of the different states or events. In the field of reliability analysis, it is of especial interest the analysis of the *faulted/down* F or *working* $\overline{F}$ (also *up* or *not faulted*) state of the equipment.

The states can be obtained from the SAN rewards that perform measurements of the marking $\mathrm{m}(pl_F) \in \{0,1\}$ of certain place pl_F that represents the failure of the equipment. The unavailability of the system $Q(t)$, that is, the probability of the system to be in the faulted state at the given time t can be approximated from the simulation results with a n_{cl}% confidence interval $CI(n_{cl})$ as (Rubinstein and Kroese, 2016):

$$\hat{Q}(t) = \frac{1}{N_{\mathrm{MC}}} \sum_{i=1}^{N_{\mathrm{MC}}} \mathrm{m}(pl_F, i, t) \tag{2.9}$$

$$S^2(t) = \frac{1}{N_{\mathrm{MC}}} \sum_{i=1}^{N_{\mathrm{MC}}} \left(\mathrm{m}(pl_F, i, t) - \hat{Q}(t)\right)^2 \tag{2.10}$$

$$CI(n_{cl}) = \left(\hat{Q}(t) \pm \frac{z_{1-\frac{\alpha}{2}} S(t)}{\sqrt{N_{\mathrm{MC}}}}\right), \alpha = 1 - \frac{n_{cl}}{100} \tag{2.11}$$

where $\mathrm{m}(pl_j, i, t)$ denotes the marking of pl_j on the MC trial number i at simulation time instant t, z_x is the x quantile of the standard normal distribution, N_{MC} is the number of MC trials performed, and S is the square root of the sample variance at the simulation time instant that the confidence interval is evaluated.

The complexity for applying the MC method is that the number of simulations performed N_{MC} must be high enough to obtain the required confidence margins that depend on the variance of the system. The smaller the variance of the samples, the fewer simulations are required to be performed to reach the required confidence margins. Normally in MC simulations *variance reduction* techniques can be applied to the sampling of the simulations, to increase precision of the resulting PDF for the same number of trials, increasing the efficiency of the method in terms of computational power required to obtain the given precision.

In the field of grid component prognostics, the goal is to estimate the Remaining Useful Life (RUL) of equipment. For dependable equipment the RUL is defined as the time remaining for the equipment for its reliability to be below its minimum acceptable reliability value $\mathcal{R}_{\mathrm{min}}$. That is:

$$\begin{aligned} &\tau_{\mathrm{RUL}} = \max \tau \in \mathbb{R}_{>0} : \mathcal{R}(t) >= \mathcal{R}_{\mathrm{min}} \text{ , } \forall t \in [0, \tau] \\ &\mathcal{R}(t) = \mathcal{P}(\tau_{\mathrm{F}} > t) \end{aligned} \tag{2.12}$$

where $\mathcal{R}(t)$ is the reliability of the system at time t and t_{F} is the Time of Failure (TOF) of the system.

When the system is not repairable after its failure, the reliability value can be obtained by calculating the system unreliability $F_F(t)$ from the failure PDF f_F, which is the unavailability Q of the system (Kumamoto and Henley, 1996):

$$\mathcal{R}(t) = 1 - F_F(t) = 1 - \int_0^t f_F(u)\mathrm{d}u = 1 - \int_0^t Q(u)\mathrm{d}u \tag{2.13}$$

which can be estimated from Equation (2.9)

2.6.1 *Monte Carlo Simulation Issues specific to the Electric Power Distribution System*

In order to obtain MC results that are representative of the actual system values, every event probability distribution, including degradation, failure, maintenance or usage patterns must be accurately represented by the N_{MC} trials. This section will address the requirements that are specific to the simulation of Electric Power Distribution faults.

Electric power distribution components are often highly reliable and their useful lives are decades long. Additionally, certain components such as circuit breakers are sometimes operated only after years of remaining in the same state (Mazzanti and Marzinotto, 2013; Megger, C. B. Megger, and Megger, 2012). Components in general degrade slowly and maintenance processes can take from hours to months. Therefore, the MC simulation of electric power distribution grids will involve the simulation of rare and correlated operational events and failure events.

See for example Figure 2.15, where a typical *loop-type* primary feeder (Gönen, 2014, Chapter 5) is employed. The grid has a Normally open (NO) *loop tie* CB cb_{LT1} that splits the grid into two sections. When any anomaly such as an over-current or a component damage in cb_{N2} causes its associated secondary system to be disconnected from its supply, the event is detected and cb_{LT1} is closed, using the other half of the loop as a redundant supply.

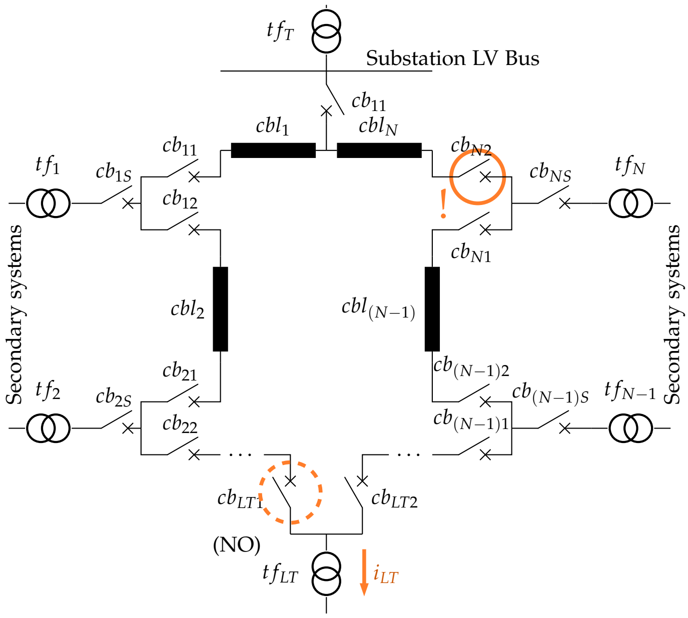


Figure 2.15: Ring topology grid fault example.

In order to model this structure, the notion of a time reference is required. Consider for example the case where a (rare) over-current in cb_{N2} causes a supply failure in tf_{LT}. After a few minutes of the failure, cb_{LT} closes, so that the supply switches to the redundant branch of the grid. When cb_{N2} recovers its closed state back, the supply switches back to the default branch by opening cb_{LT1}.

This event causes a switching sequence that affects the cable degradation rate by varying its load condition. The switching events in cb_{N2} are rare, but once the switching occurs, it causes an immediate reaction on cb_{LT1} .

Unlike some failure probability analysis methods that classify events in depending on the order or the time intervals that the events occur (Allen, 1983; Walker and Papadopoulos, 2009), the notion of a more precise absolute time reference is required to model electric power distributions like the one shown in Figure 2.15.

Generally, components in the grid, such as cables or transformers, do not fail in several decades. The over-current events are rare, and they will occur with large intervals (months or even years) and small λ_{Fault} fault rates which will be generated by the stochastic processes governing it. However, the operational switching of cb_{LT1} is governed by a stochastic (or deterministic) process that exhibits high λ_{op} rates that can represent short intervals of a few minutes. At these intervals, failures may occur both on cb_{LT1} due to the switching, or the both on the CBs and cables on the redundant branch, due to the over-current that the switching caused.

In this context, the correct representation of the PDF for those fast operational events is generally not an issue, given that their variances will be relatively small in comparison to the rare events, and fewer MC trials are required to accurately represent them. However, the operational events are often the consequence of other rare occurrences, causing it to be a rare event.

The rare operational events can also be the result of fast switching or cascading events in the power distribution grid (Dueñas-Osorio and Vemuru, 2009). The simulation of these short time intervals may have no effect on the maintenance processes due to their relatively negligible short intervals. However, the correct simulation of the event intervals can be critical to represent the sequencing of certain events that may rarely lead to failures. Therefore, it is important to consider the representation of fast switching events whose combination may have an effect on the final system unavailability that may be uncaught if these simulation intervals are neglected.

The requirement to simulate fast operational event increases the requirement for the MC time resolution, while the long failure times and rare events require long mission times to be simulated as it will be discussed in Chapter 5

2.7 Model Considerations and Scope Of This Thesis

The target of this thesis is permitting the prognosis of underground cables associated to a grid. In this context, a model of the grid is integrated in the form of a HSAN MAC. SAN are a very general formalism that covers a range of applications. While the methods developed for the SAN simulation are generalist, the implementations had to be limited to the scope of the thesis, due to resource requirements and time-frames. In the context of electric power distribution grids, only part of the SAN features are required, including:

1. Input and output gate functions permit describing complex physical behaviors of components.

2. Both deterministic and stochastic time intervals are required to describe component switching, and predicate functions permit describing failure conditions.

3. Instantaneous activities permit describing the different possible outcomes when the outcome of an activity is random.

4. Rewards are required to obtain the statistical measurements out of the model.

The following features have not been considered, given that they found to be necessary for the general description of the grid behavior:

1. The reactivation mechanism of activities has not been considered, because the mechanism can be represented by extra timed activities and it was found not to be necessary for the grid models used through this thesis.

2. Cases have not been considered, because as described by (Movaghar, 2001), they can be replaced by extra instantaneous activities.

An accurate representation of the grid requires full understanding of the degradation processes of all the components involved. The integration of physical models for each of the components has been described. However, the degradation process of many of the components that compose the grid are still subject of recent studies, and the prognosis models of components such as CB or transformers are still evolving (J. I. Aizpurua, Catterson, Papadopoulos, et al., 2017; Risos and Gouws, 2019; J. I. Aizpurua, McArthur, et al., 2019).

Often, the failure of these components are assumed to be either exponentially or Weibull distributed (Amare, Helvik, and Heegaard, 2018; Suwanasri et al., 2014). The limitation of these models is that the future behavior and PDF of the components may vary with usage, and cannot be considered to be uncorrelated to other events in the grid. The prognosis will still be valid while the PDF obtained from previous Condition Monitoring (CM) holds valid. Longer prognosis of the grid will, however, require more accurate component models. Throughout this thesis, only cables are modeled by an accurate physical model, while the events of the grid are modeled to follow known PDFs.

3

CONDITION MONITORING AND PROGNOSTICS OF UNDERGROUND CABLES AND OTHER GRID EQUIPMENT

The prognosis of the equipment usually involves two phases shown in Figure 3.1 (Kim, An, and Choi, 2017b):

- The condition monitoring and diagnostics process that estimates the current equipment degradation or failure probability level.
- Prediction of the future health-state of the system.

The Condition Monitoring (CM) and current health-state estimation (or diagnostics) involves monitoring the equipment by performing either direct measurements or estimation by indirect parameter measurements of the component degradation up to now.

The prognosis phase analyzes the evolution of the present state of equipment condition under the predicted conditions (see Figure 3.1). This prediction is usually based on feature-based or model-based analysis of the equipment health state (Kim, An, and Choi, 2017b). The evolution of the equipment is estimated until its health-state crosses the acceptable limit, e.g., an insulator degradation level or minimum reliability level of the equipment. The time instant when the crossing of the threshold occurs defines the End Of Life (EOL) of the equipment and the remaining time-period until this time instant is the Remaining Useful Life (RUL).

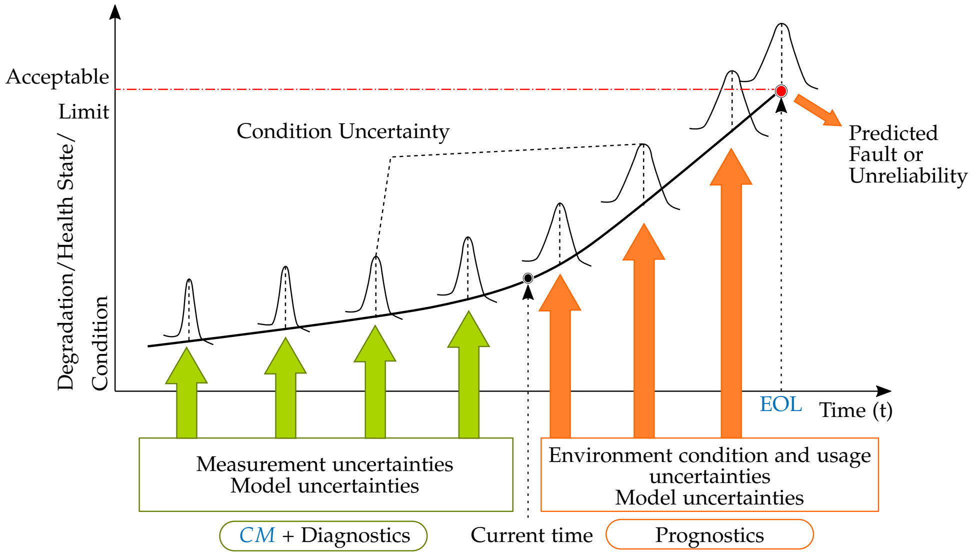


Figure 3.1: CM and diagnostics, and prognosis phase uncertainties

This chapter describes the models and methods employed for monitoring underground cables, and discusses its considerations and limitations.

3.1 Equipment Condition Monitoring Methods

The data acquisition and CM often involves a variety of complex problems to estimate the current condition of the component. Model uncertainties, measurement noise, varying parameter estimations or feature extraction mechanisms can cause errors on the estimated values.

A variety of techniques, often data-driven or hybrid, have been used in the literature for the CM of the component, such as Unscented Kalman Filters (UKF) (X. Zhang and Pisu, 2012); Particle Filter (PF) that are Sequential Monte Carlo (SMC) based method that predicts the future Probability Density Function (PDF) from by recursively propagating a few samples (particles) of the original distribution (Arulampalam et al., 2002; M. Orchard, Tobar, and G. Vachtsevanos, 2009; M. E. Orchard and G. J. Vachtsevanos, 2009; Zhou et al., 2009; Engel et al., 2000); Neural networks based methods like Adaptive Recurrent Neural Networks (ARNN) (J. Liu et al., 2010), robust autoassociative neural networks (Kramer, 1992) and Dynamic Wavelet Neural Networks (DWNN) (P. Wang and G. Vachtsevanos, 2001); Fuzzy Logic (Zio and Di Maio, 2010); and Regression of degradation estimates (S. Cheng and Pecht, 2007).

However, if the equipment is expected to be subjected to different usage and/or environmental conditions in the future, the CM data cannot be directly extrapolated or projected to the future behavior of the equipment. For example, varying temperatures or load will cause varying degradation patterns on the cables.

In this context, it is possible to use physics-of-failure based methods, i.e. model-based methods, to diagnose the current degradation condition of the equipment, including underground cables (Zhong et al., 2015; Mariut et al., 2012).

Additionally, if such a physics-of-failure model of a component exists, e.g. a degradation model of a cable, this model can also be applied to predict the Remaining Useful Life (RUL) of the equipment, if the future operation conditions can be estimated. That is, the physics-based model used for the diagnostics of the cable can be applied to the cable prognostics analysis.

As discussed in Section 1.3.1, the degradation process of an underground cable can be estimated from the thermal transient analysis of the cable insulator (Mazzanti, 2007). However, the process requires considering a full transient analysis of a varying load system, rather than a steady-state thermal analysis, and the process is affected by multiple sources of uncertainty.

So as to consider a full transient analysis, along with the different sources of uncertainty for the CM and prognosis phases, this thesis adopts a model-based approach.

3.2 UNCERTAINTIES IN THE MONITORING PROCESS OF UNDERGROUND CABLES

There are different sources of uncertainty inherent in the CM and prognosis of any equipment (Diaz-Aguiló and León, 2015; Pan, Z. Wang, and Sun, 2015; Sankararaman, 2015; Ekwaro-Osire, Gonçalves, and Alemayehu, 2017). For example, measurements can be subjected to errors or calibration biases of the sensors, certain parameters of the models are frequently assumed to be constant for simplicity or estimated indirectly, and the models themselves may be incomplete or inaccurate due to the lack of knowledge.

In the case of the model-based monitoring of underground cables, the load measurements will be affected by noise and offset/bias of the current sensors (Petercem Sensors, 2018) and the thermal models of the cables will be affected by varying environment parameters such as soil diffusivity or soil temperatures (Diaz-Aguiló and León, 2015). These uncertain environmental conditions and model errors can cause the cables to be subjected to insulation temperatures that deviate from the expected values.

As a result of this temperature deviation, the cable insulation will follow unexpected degradation patterns, leading to an incorrect estimation of the current condition of the cable. The impact of the errors in the estimation of the degradation in absolute terms is dependent on the time-frame of the condition monitoring, given the cumulative nature of the degradation models (Sánchez-Silva and Klutke, 2016a, Chapter 4), and the accuracy of the predicted future degradation will depend on the accuracy of the expected conditions and the model accuracy.

Accordingly, it is necessary to model the uncertainties, evaluate their impact on the degradation models, and to obtain probabilistic results for the current condition of the equipment, considering the degradation monitoring a stochastic process (see Figure 3.1). Due to the complexity of evaluating analytically the PDF of the resulting degradation models, these results are usually obtained by performing Monte Carlo (MC) simulations of the physical models (G. J. Anders, 2005, Chapter 4).

3.3 Model-Based Condition Monitoring and Prognostics of Cables

The electrical insulation is considered to be one of the most critical failure points in high-voltage underground cables (Robles et al., 2016; Hyvönen et al., 2008). Unlike aerial cables, underground cables consist of multiple layers that act as thermal barriers with a given thermal resistivity and capacitance. Cross-Linked Polyethylene (XLPE) cables (see Figure 1.1) generally consist of a main conductor, shield, the insulator layer, screen, sheath, armoring, and the outer cover or jacket (Ametani, Ohno, and Nagaoka, 2015, Chapter 1).

This thesis concentrates on the analysis of the thermo-electrical erosion process of the underground cable insulators. The analysis is based on a model-based CM and subsequent model-based prognostics. The models impose structure and environmental assumptions that lead to their formulation. The following sections describe the methods and assumptions that are used through this thesis for the thermo-electrical analysis of the underground cable, which are applied in the Field Programmable Gate Array (FPGA) architecture described in Chapter 5.

3.3.1 *Thermal Model of Underground Cables*

The thermal heat transfer process for any underground cable can be described by the Fourier thermal diffusivity equation. In the case of underground cables, the heating of the cable depends on the power dissipated both within the cable under analysis and within the adjacent cables.

Throughout this thesis the length of the cable segments under analysis is considered to be infinite in comparison to their diameter, and uniform characteristics across the cable, for thermal analysis purposes. In this case, a two-dimensional differential equation is obtained. Considering a constant dissipated power density W in the cable, the net energy accumulated within the material (the gradient of the energy flow) causes a temperature rise that can be described by the differential equation in cylindrical coordinates (Hahn and Özişik, 2012)

$$\frac{1}{r}\frac{\partial}{\partial r}\left(\frac{1}{\rho}r\frac{\partial \theta}{\partial r}\right) + W = c\frac{\partial \theta}{\partial t} \tag{3.1}$$

where W is the constant dissipated power density, ϱ is the thermal resistivity of the material, c is the volumetric thermal capacity of the material, r is the radius and θ is the temperature.

3.3.1.1 *Sheath and Armor Losses*

In addition to the current circulating in the main conductor, currents are induced in the metallic sheath and armor that generate extra heating. Currents in the sheath are generated due to (a) eddy currents generated by the conductor current itself, (b) eddy currents generated by neighboring cables, and (c) currents generated due to circulating currents caused by the bonding at both ends. In the armor, currents are generated by (a) circulating currents due to bonding on both ends, and (b) caused by hysteresis in the case of magnetic armors. The circulating currents can be canceled by cross-connecting the sheath of connected cable sections (Ametani, Ohno, and Nagaoka, 2015, Section 1.4.2.3) as shown in Figure 3.2.

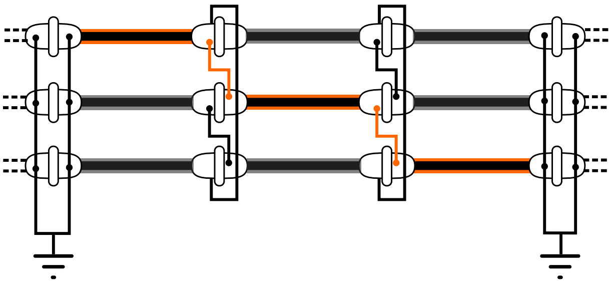


Figure 3.2: Cross bonded cable sheath diagram.

If the current loads in the three phases are balanced, the power losses on the sheath or screen can be calculated as proportional to the load current by a loss factor λ_{sh} (commonly denominated λ_1 in the IEC60287-1-1 standard), and a loss factor of λ_{a} for the armor (denominated λ_2 in the standard), as (IEC, 2006)

$$W_{\mathrm{sh}} = \lambda_{\mathrm{sh}} W_{\mathrm{c}} \tag{3.2}$$

$$W_{\mathrm{a}} = \lambda_{\mathrm{a}} W_{\mathrm{c}} \tag{3.3}$$

$$\lambda_{\mathrm{sh}} = \lambda'_{\mathrm{sh}} + \lambda''_{\mathrm{sh}} \tag{3.4}$$

$$\lambda_{\mathrm{a}} = \lambda'_{\mathrm{a}} + \lambda''_{\mathrm{a}} \tag{3.5}$$

where W_{c} is the power loss density in the main conductor, W_{sh} is the power loss density in the sheath, W_{a} is the power loss density in the armor, λ'_{sh} and λ'_{a} are the loss factors due to the circulating currents in the sheath and the armor, λ''_{sh} is the loss factor due to the eddy currents, and λ''_{a} is the loss factor due to the hysteresis in a magnetic armor.

Methods for the calculation of the loss factors different cable set structures can be found in (IEC, 2006 ; G. J. Anders, 2005, Chapter 1).

3.3.1.2 *Definition of the Thermal RC Ladder Model for an Insulator*

The temperature along the radius r of each cylindrical insulator layer i, as shown in Figure 3.3, with thermal resistivity ϱ_i subjected to a constant dissipated power W per unit length, and assuming a negligible thermal capacity of the inner conductor, is non-linearly distributed as (Bejan and Kraus, 2003, Section 3.4.2)

$$W = \frac{2\pi(\theta_{\text{in}} - \theta_{\text{out}})}{\varrho_i \ln\left(\frac{r_{i,\text{out}}}{r_{i,\text{in}}}\right)} \tag{3.6}$$

$$\theta(r) = \theta_{\text{in}} + \frac{\theta_{\text{in}} - \theta_{\text{out}}}{\ln\left(\frac{r_{i,\text{in}}}{r_{i,\text{out}}}\right)} \ln\left(\frac{r}{r_{i,\text{in}}}\right) \tag{3.7}$$

where $r_{i,\text{in}}$ and $r_{i,\text{out}}$ are the inner and outer radius of the layer i, and θ_{in} and θ_{out} are the inner radius temperature and outer radius temperatures respectively. Therefore, for cylindrical cable segments, the thermal resistance per unit length $\mathcal{T}_i$ of a cable layer i can be calculated as:

$$\mathcal{T}_i = \frac{\theta_{\text{in}} - \theta_{\text{out}}}{W} = \frac{\varrho_i}{2\pi} \ln \frac{r_{i,\text{out}}}{r_{i,\text{in}}} \tag{3.8}$$

where ϱ_i is thermal resistivity of the material in layer i.

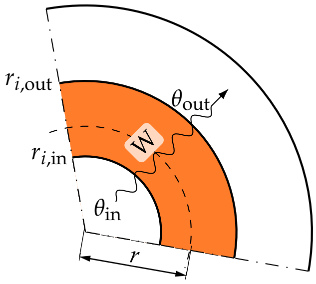


Figure 3.3: Section of cable insulator.

Substituting θ_{out} from Equation (3.6) into Equation (3.7)

$$\theta(r) = \theta_{\text{in}} - \frac{\varrho_i}{2\pi} W \ln\left(\frac{r}{r_{i,\text{in}}}\right) \tag{3.9}$$

Therefore, the temperature is logarithmically distributed along the insulator and the temperature gradient is proportional to the heat flow.

Taking into account the temperature gradients within the cable body when solving the transient temperature responses for the differential equation in Equation (3.1) can be complex, but the solution can be approximated by splitting the insulation layer into multiple sub-layers employing Van Wormer coefficients (G. J. Anders, 1997, Chapter3 ; Van Wormer, 1955; Millar, 2006).

If the thermal conductivity of a layer j were relatively high and the temperature could be considered to be uniform through the layer, an equivalent capacitance per unit length C_j of the layer could be defined for the thermal calculations as:

$$C_j = \pi(r_{j,\text{out}}^2 - r_{j,\text{in}}^2)c_j \tag{3.10}$$

where c_j is the volumetric specific heat of the material in layer j, and $r_{j,\text{out}}$ and $r_{j,\text{in}}$ are its outer radius and its inner radius respectively.

In the case of cable insulators, as shown by (Van Wormer, 1955), any insulator i with $\mathcal{T}_i$ thermal resistance, C_i thermal capacitance, c_i volumetric specific heat and V_i cross-section (volume of insulator per unit length) the cylindrical layer can be split into the sum of two equivalent thermal capacitances C_i' and C_i'', split with a factor p_i as

$$C_i = \underbrace{p_i V_i c_i}_{C_i'} \theta_{\text{in}} + \underbrace{(1-p_i) V_i c_i}_{C_i''} \theta_{\text{out}} = c_i \int_{r_{i,\text{in}}}^{r_{i,\text{out}}} \theta(r) 2\pi r \mathrm{d}r \tag{3.11}$$

with $V_i = \pi(r_{i,\text{out}}^2 - r_{i,\text{in}}^2)$

Substituting $\theta(r)$ with Equation (3.7), we obtain

$$p_i c_i \theta_c \pi \left(r_{i,\text{out}}^2 - r_{i,\text{in}}^2\right) + (1-p_i) c_i \theta_s \pi \left(r_{i,\text{out}}^2 - r_{i,\text{in}}^2\right) =$$

$$c_i \theta_{\text{in}} \pi (r_{i,\text{out}}^2 - r_{i,\text{in}}^2) + \frac{2\pi c_i \left(\theta_{\text{in}} - \theta_{\text{out}}\right)}{\ln\left(\frac{r_{i,\text{in}}}{r_{i,\text{out}}}\right)} \left[-\frac{r_{i,\text{out}}^2 - r_{i,\text{in}}^2}{4} + \frac{r_{i,\text{out}}^2 \ln\left(\frac{r_{i,\text{out}}}{r_{i,\text{in}}}\right)}{2} \right]$$

that can be simplified as in Appendix A.5

$$p_i = p(r_{i,\text{out}}, r_{i,\text{in}}) = \frac{1}{2\ln\left(\frac{r_{i,\text{out}}}{r_{i,\text{in}}}\right)} - \frac{1}{\left(\frac{r_{i,\text{out}}}{r_{i,\text{in}}}\right)^2 - 1} \tag{3.12}$$

For greater accuracy when modeling on small cable conductors covered with large insulators or short transient analysis, it is suggested that this insulator is divided into two separate insulator sub-layers i1 and i2 with equal thermal resistivity $\frac{\mathcal{T}_i}{2}$, and thermal capacitances per unit length $C_{\text{i}1}$ and $C_{\text{i}2}$ respectively. It is generally assumed that a transition is short if it lasts less than $\frac{1}{3}\sum_i \mathcal{T}_i \sum_j C_j$ (Van Wormer, 1955, G. J. Anders, 2005, Section 1.3.3.1).

Each sub-layer is then split into two thermal capacitances with the procedure described above. The segmentation is displayed in Figure 3.4, where $\theta_{\text{in}} = \theta_{\text{c}}$ is the conductor temperature and $\theta_{\text{out}} = \theta_{\text{sh}}$ is the sheath temperature. The equivalent thermal resistance of each part at a given radius $r_{½}$ will be $\mathcal{T}_{r_{½}} = \frac{\mathcal{T}_i}{2}$, being this radius

$$\mathcal{T}_{r_{½}} = \frac{\mathcal{T}_i}{2} = \frac{\theta_{\text{in}} - \theta(r_{½})}{W} = \frac{\varrho_i}{2\pi} \ln\left(\frac{r_{½}}{r_{i,\text{in}}}\right) \therefore r_{½} = \sqrt{r_{i,\text{out}} r_{i,\text{in}}} \tag{3.13}$$

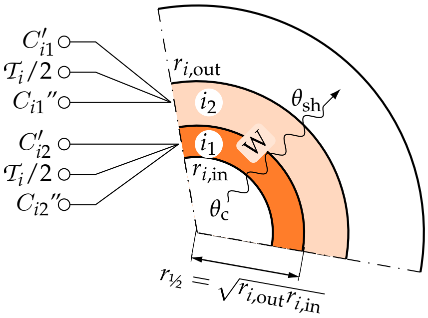


Figure 3.4: Van Wormer T model splitting for cables with larger insulators.

Each of the resulting segments is then represented by separate thermal capacities employing the Van Wormer method. The resultant new Van Wormer coefficients for each segment, p_{i1} and p_{i2} can be calculated similarly, and result in an equally valued p_i^* factor:

$$p_{\mathrm{i1}} = p(r_{½}, r_{i,\mathrm{in}}) = \frac{1}{2\ln\left(\frac{r_{½}}{r_{i,\mathrm{in}}}\right)} - \frac{1}{\left(\frac{r_{½}}{r_{i,\mathrm{in}}}\right)^2 - 1} = \frac{1}{\ln\frac{r_{i,\mathrm{out}}}{r_{i,\mathrm{in}}}} - \frac{1}{\frac{r_{i,\mathrm{out}}}{r_{i,\mathrm{in}}} - 1} \tag{3.14}$$

$$p_{\mathrm{i2}} = p_{\mathrm{i1}} = p_i^* \tag{3.15}$$

Being both coefficients equal valued, the term p_i^* will be employed for both in the remainder of this document. The thermal capacitances per unit length of the resulting equivalent circuit are calculated as:

$$\begin{aligned}
C'_{\mathrm{i1}} &= p_{\mathrm{i}}^*\, C_{\mathrm{i1}} = p_{\mathrm{i}}^*\, \pi \left(r_{½}^2 - r_{\mathrm{c}}^2\right) c_{\mathrm{i}} = p_{\mathrm{i}}^* \pi \left(r_{\mathrm{i}} r_{\mathrm{c}} - r_{\mathrm{c}}^2\right) c_{\mathrm{i}} \\
C''_{\mathrm{i1}} &= (1-p_{\mathrm{i}}^*) C_{\mathrm{i1}} = (1-p_{\mathrm{i}}^*) \pi \left(r_{\mathrm{i}} r_{\mathrm{c}} - r_{\mathrm{c}}^2\right) c_{\mathrm{i}} \\
C'_{\mathrm{i2}} &= p_{\mathrm{i}}^*\, C_{\mathrm{i2}} = p_{\mathrm{i}}^*\, \pi \left(r_{\mathrm{i}}^2 - r_{\mathrm{i}} r_{\mathrm{c}}\right) c_{\mathrm{i}} \\
C''_{\mathrm{i2}} &= (1-p_{\mathrm{i}}^*) C_{\mathrm{i2}} = (1-p_{\mathrm{i}}^*) \pi \left(r_{\mathrm{i}}^2 - r_{\mathrm{i}} r_{\mathrm{c}}\right) c_{\mathrm{i}}
\end{aligned}$$

where r_{c} is the conductor radius and r_{i} is the outer radius of the insulator. The resulting equivalent circuit is Figure 3.5, being $\mathcal{T}_i$ is the insulator thermal resistance per unit length, C_{c} is the thermal capacitance of the conductor per unit length, and C_{sh} is the capacitance per unit length of the next layer in its surface (generally the sheath).

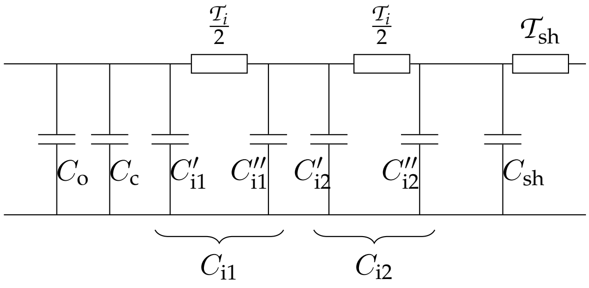


Figure 3.5: Van Wormer equivalent of a cylindrical insulator in between the conductor and its surface equivalent circuit

3.3.1.3 *Equivalent RC Model of the Cable Self-Heating Process*

The self-heating process of a full XLPE cable is modeled using an equivalent RC ladder circuit representing each of the cable insulator and conductor layers of the cable, as shown in Figure 3.6 (G. Anders and El-Kady, 1992). In this model, the energy dissipated in the different x layers is represented as W_x power sources, and each of the material layers is transformed into a set of a thermal resistances $\mathcal{T}_x$ and thermal capacitances C_x that connect to the outside of the cable.

The capacitance and resistances representing the sub-layers generated by the Van Wormer coefficients in Figure 3.6 are named C_{xj} and $\frac{\mathcal{T}_x}{2}$, being x the layer name, and j the sub-layer number.

Considering the cable to feature uniform characteristics throughout its total length, W_x are defined as power dissipated per unit of cable length ($\mathrm{W\,m^{-1}}$), $\mathcal{T}_x$ as thermal resistance per unit length ($\mathrm{K\,m\,W^{-1}}$), and C_x as capacitance per unit length ($\mathrm{J\,K^{-1}\,m^{-1}}$).

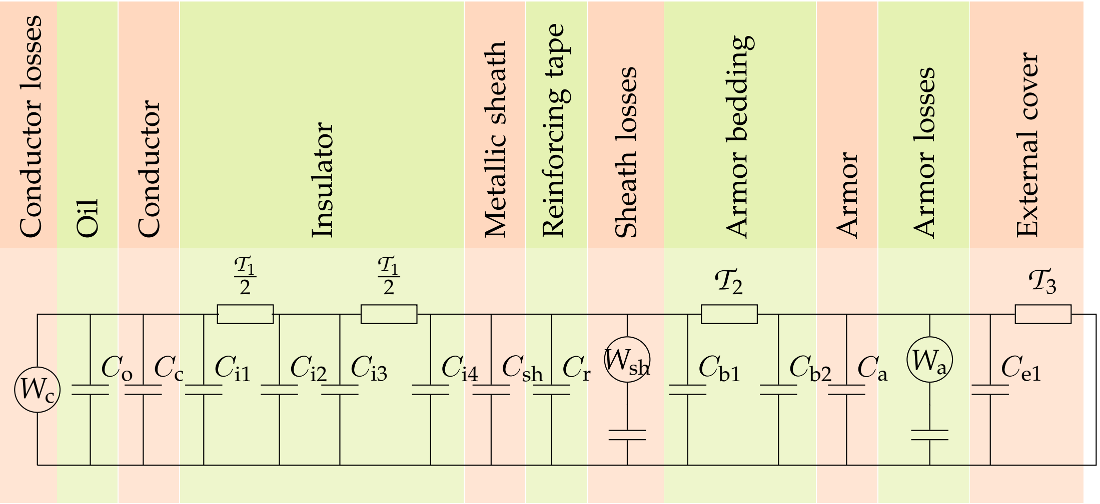


Figure 3.6: Self-heating thermal model equivalent ladder circuit.

3.3.1.4 *Construction of the equivalent two loop circuit*

The losses on the metallic sheath W_{sh} and the loses in the armor W_{a} are directly proportional to the losses in the main conductor W_{c} (see Section 3.3.1.1). Thus, a circuit equivalent to the one in Figure 3.6 can be obtained in the general form shown in Figure 3.7.

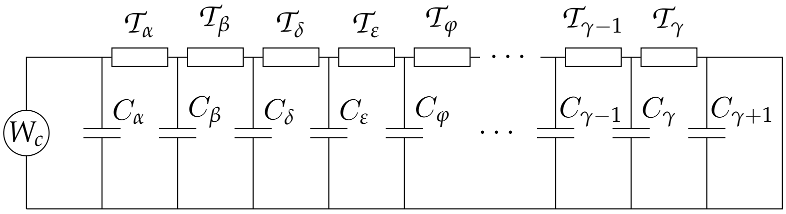


Figure 3.7: Equivalent self-heating thermal model.

In order to account for the sheath losses, the thermal resistances in Figure 3.7 beyond the sheath are multiplied by the corresponding loss factor $(1+\lambda_{\mathrm{sh}})$, and its own capacity and the ones beyond divided by the same factor. The same method is applied for accounting for the armor losses. Therefore, adapting the circuit in Figure 3.6 to the general form in Figure 3.7 results in (G. J. Anders, 2005, Section 1.3.3.2).

$$C_\alpha = C_{\mathrm{c}} + C_{\mathrm{o}} + C_{\mathrm{i1'}} \tag{3.16}$$

$$C_\beta = C_{\mathrm{i1''}} + C_{\mathrm{i2'}} \tag{3.17}$$

$$C_\delta = C_{\mathrm{i2''}} + \frac{1}{1+\lambda_{\mathrm{sh}}}\left(C_{\mathrm{sh}} + C_{\mathrm{r}} + C_{\mathrm{b1}}\right) \tag{3.18}$$

$$C_\varepsilon = C_{\mathrm{b2}} + \frac{1}{1+\lambda_{\mathrm{sh}}+\lambda_{\mathrm{a}}} C_{\mathrm{a}} \tag{3.19}$$

$$C_\gamma = \frac{1}{1+\lambda_{\mathrm{sh}}+\lambda_{\mathrm{a}}} C_{\mathrm{e1}} \tag{3.20}$$

$$\mathcal{T}_\alpha = \frac{\mathcal{T}_1}{2} \tag{3.21}$$

$$\mathcal{T}_\beta = \frac{\mathcal{T}_1}{2} \tag{3.22}$$

$$\mathcal{T}_\delta = (1+\lambda_{\mathrm{sh}})\,\mathcal{T}_2 \tag{3.23}$$

$$\mathcal{T}_\gamma = (1+\lambda_{\mathrm{sh}}+\lambda_{\mathrm{a}})\,\mathcal{T}_3 \tag{3.24}$$

While obtaining a general transfer function of the response of the circuit in Figure 3.7 is possible, it has been demonstrated that the circuit can be approximated by a two loop circuit shown in Figure 3.8 (IEC, 2008, G. J. Anders, 1997, Chapter 3). where

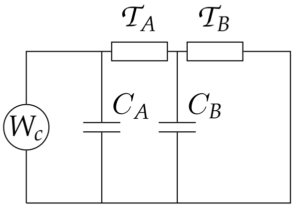


Figure 3.8: Approximate two-loop circuit of self-heating thermal model.

$$C_A = C_\alpha \tag{3.25}$$
$$\mathcal{T}_A = \mathcal{T}_\alpha \tag{3.26}$$
$$C_B = C_\beta + \left(\frac{\mathcal{T}_\delta + \mathcal{T}_\epsilon + \cdots + \mathcal{T}_\gamma}{\mathcal{T}_\beta + \mathcal{T}_\delta + \mathcal{T}_\epsilon + \cdots + \mathcal{T}_\gamma}\right)^2 C_\delta + \left(\frac{\mathcal{T}_\epsilon + \cdots + \mathcal{T}_\gamma}{\mathcal{T}_\beta + \mathcal{T}_\delta + \mathcal{T}_\epsilon + \cdots + \mathcal{T}_\gamma}\right)^2 C_\epsilon + \cdots + \left(\frac{\mathcal{T}_\gamma}{\mathcal{T}_\beta + \mathcal{T}_\delta + \mathcal{T}_\epsilon + \cdots + \mathcal{T}_\gamma}\right)^2 C_\gamma \tag{3.27}$$
$$\mathcal{T}_B = \mathcal{T}_\beta + \mathcal{T}_\delta + \mathcal{T}_\epsilon + \cdots + \mathcal{T}_\gamma \tag{3.28}$$

The thermal response of the reduced model can then be represented by a second order Laplace transfer function $H(s)$ (G. J. Anders, 2005, Chapter 1).

$$\mathcal{L}\{\theta_s(t)\} = \theta_s(s) = H(s)W_c(s) \tag{3.29}$$
$$H(s) \approx \frac{\mathcal{T}_A + \mathcal{T}_B + (C_B\mathcal{T}_A\mathcal{T}_B)s}{C_AC_B\mathcal{T}_A\mathcal{T}_Bs^2 + (C_A(\mathcal{T}_A + \mathcal{T}_B) + C_B\mathcal{T}_B)s + 1} \tag{3.30}$$

where $W_c(s)$ is the Laplace transform of the power dissipated in the main cable conductor, and $\mathcal{T}_A$, $\mathcal{T}_B$, C_A and C_B are the thermal resistance and capacitances of the equivalent thermal circuit in Figure 3.8.

In addition to the self-heating process caused by the power dissipated within the cable under analysis, two other effects need to be considered when modeling the temperature rise of the cable:

1. The cable is buried underground. Therefore, the conductivity and diffusivity of the soil affect the dissipation of the heat to the air.
2. Nearby cables can cause mutual heating and an additional temperature rise.
3. When the cable is laid in a pipe with either natural or forced air convection, the air flow causes heat flux and cooling of the cable.

The thermal model used in this thesis considers both the diffusion process of the soil and the mutual cable heating. The convection effect has not been employed in the models through this thesis, although the effect of air convection can be included in the developed models as described in (G. J. Anders, 1997, Chapter 3 ; Makhkamova, 2011). The thermal model employed in this thesis is the one adopted in IEC60853-2 (IEC, 2008) and it has been accepted and validated by several authors (G. Anders and El-Kady, 1992; Bicen, 2017; Stojanovic, Tasic, and Ristic, 2013) for modeling the temperature evolution of the cable insulator. In this model, the thermal temperature rise $\theta(t)$ above ambient temperature is calculated as (IEC, 2008)

$$\theta(t) = \theta_s(t) + \zeta(t)\theta_e(t) + \zeta(t)\sum_k \theta_{pk}(t) + \theta_{pdk}(t) \tag{3.31}$$

where $\theta_s(t)$ is self-heating process temperature rise caused by the total power dissipated within the cable of interest, described by Equation (3.29). $\theta_e(t)$ is the temperature rise caused by influence of the soil, $\theta_{pk}(t)$ and $\theta_{pdk}(t)$ are the temperature rises caused on the cable of interest p by the power dissipated in each adjacent cable k and their dielectric losses respectively, and $\zeta(t)$ is the *attainment factor* for the transient temperature rise between the conductor and the outer surface, as described below (G. J. Anders, 2005, Chapter 1). The effect of mutual heating caused by dielectric losses in adjacent cables $\theta_{pdk}(t)$ can be generally neglected for distribution networks and therefore it has not been included throughout this thesis document (Aras and Biçen, 2013).

If the cable is considered an infinite line buried in an infinite soil medium with uniform thermal characteristics, the solution to Equation (3.1) for the transient thermal response $\theta_e(t)$ to a constant load step W_{tot} due to the effect of the soil can be solved as (Carslaw and Jaeger, 1959, Chapter 10 ; Bejan and Kraus, 2003 ; G. J. Anders, 1997, Chapter 5)

$$\theta_{\text{e}}(t) = W_{\text{tot}} \frac{\varrho_{\text{s}}}{4\pi} \left[-\text{Ei}\left(-\frac{d^2}{4\delta_{\text{s}} t} \right) \right] \tag{3.32}$$

$$\delta_{\text{s}} = \frac{1}{\varrho_{\text{s}} c_{\text{s}}} \tag{3.33}$$

where d is the distance from the cable, ϱ_{s} is the thermal resistivity of the soil , δ_{s} is the thermal diffusivity of the soil, c_{s} is the volumetric thermal capacity of the soil, and $\text{Ei}\left(\cdot\right)$ is the exponential integral function, defined for real negative values of its argument (See Appendix A.7 for the description of the function and its properties).

However, if the soil medium is considered limited by the soil surface that is considered isothermal, the solution of Equation (3.1) for the transient thermal response to a load step due to the effect of the soil can be calculated by the Kennelly hypothesis (Kennelly, 1893)[1] as described in IEC60853-2 (G. J. Anders, 1997, Chapter 5 ; IEC, 2008).

By this model, the heating caused at any point P by a cable p laid underground at a depth L_d and emitting a W_{tot} power is equal to the effect of the heating caused by the cable in an infinite medium added to the effect of a *mirror* cable above the soil as shown in Figure 3.9

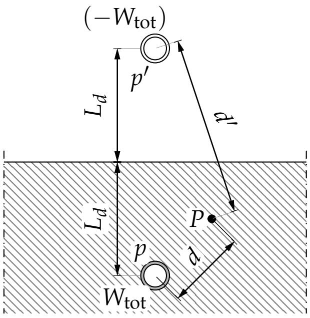


Figure 3.9: Kennelly hypothesis.

---

[1] Much to my regret, I am citing this article indirectly. As also reported by other authors (Millar, 2006), I was unable to locate the original article due to its age.

If the point P is placed exactly at the top of the cable, at the outer diameter position, being the cable diameter D_e and therefore the distance $d = \frac{D_\mathrm{e}}{2}$, and $d' = 2L_d - \frac{D_\mathrm{e}}{2} \approx 2L_d$

$$\theta_\mathrm{e}(t) = W_\mathrm{tot} \frac{\varrho_\mathrm{s}}{4\pi} \left[-\mathrm{Ei}\left(-\frac{\left(\frac{D_\mathrm{e}}{2}\right)^2}{4\delta_s t} \right) \right] + (-W_\mathrm{tot}) \frac{\rho_s}{4\pi} \left[-\mathrm{Ei}\left(-\frac{\left(2L_d - \frac{D_\mathrm{e}}{2}\right)^2}{4\delta_s t} \right) \right] \tag{3.34}$$

$$\approx W_\mathrm{tot} \frac{\varrho_\mathrm{s}}{4\pi} \left[-\mathrm{Ei}\left(-\frac{D_e^2}{16\delta_\mathrm{s} t} \right) + \mathrm{Ei}\left(-\frac{L_d^2}{\delta_\mathrm{s} t} \right) \right] \tag{3.35}$$

where W_tot is the total power dissipated in the cable, including sheath and armor losses, ϱ_s and δ_s are the thermal resistivity and diffusivity of the soil, D_e the external cable diameter, and L_d is the depth at which the cable is laid.

Finally, the thermal response to a mutual heating caused by each adjacent cable k on the cable p can be calculated by applying again the Kennelly hypothesis:

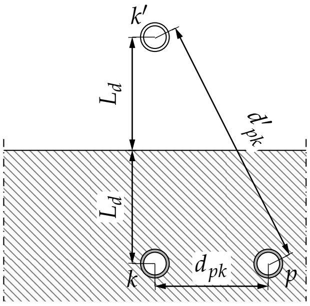


Figure 3.10: Mutual heating between two cables.

$$\theta_{pk}(t) = W_\mathrm{tot} \frac{\varrho_\mathrm{s}}{4\pi} \left[-\mathrm{Ei}\left(-\frac{d_{pk}^2}{4\delta_\mathrm{s} t} \right) + \mathrm{Ei}\left(-\frac{d'^2_{pk}}{4\delta_\mathrm{s} t} \right) \right] \tag{3.36}$$

where d_{pk} and d'_{pk} are the distance to the adjacent cable and its image respectively.

As it can be observed in Figure 3.11, part of the total heat generated in the cable, including conductor, sheath, and armor losses, referred to as W_tot in Equation (3.35) and Equation (3.36), does not get propagated to the soil and it is accumulated within the cable, rising its different layer temperatures. Current standards and literature suggest estimating the power loss by an additional *attainment factor* $\zeta(t)$. This factor is applied both to the soil effect and mutual-heating process models as a time-varying multiplier factor in Equation (3.31).

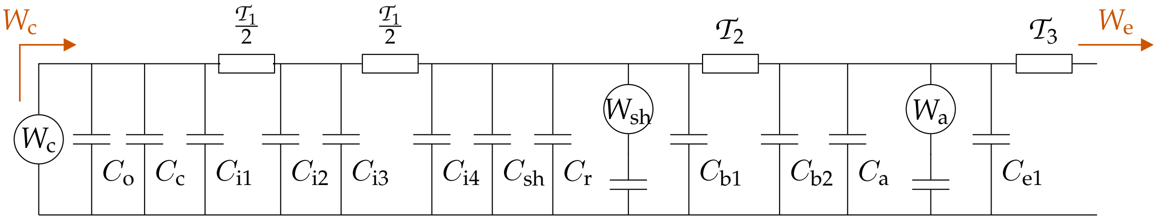


Figure 3.11: Heat accumulated in the various cable layers.

Accordingly, for a N_φ single-phase underground cable set as shown in Figure 3.12, the total temperature difference between the air and the p^th cable insulator can be estimated as

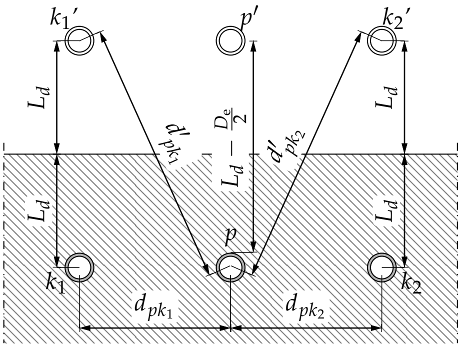


Figure 3.12: Soil effect and mutual heating calculation for $N_\varphi = 3$ single-phase underground cable set.

$$\begin{aligned}\theta(t) &= \theta_s(t) + \zeta(t)\theta_e(t) + \zeta(t)\sum_{k\in\mathcal{S}} \theta_{pk}(t) + \theta_{pdk}(t) \\ \mathcal{S} &= \{1, 2, \dots p-1, p+1, \dots N_\varphi\}\end{aligned} \tag{3.37}$$

For the case of equal total power W_{tot} dissipating underground cables, and neglecting the effect of the mutual heating due to the dielectric losses as before,

$$\begin{aligned}\theta(t) =& \theta_s(t) + \zeta(t)\cdot W_{\text{tot}}\frac{\varrho_s}{4\pi}\left\{\left[-\text{Ei}\left(-\frac{D_e^2}{16\delta_s t}\right) + \text{Ei}\left(-\frac{L_d^2}{\delta_s t}\right)\right]\right. \\ &\left. + \sum_{k\in\mathcal{S}}\left[-\text{Ei}\left(-\frac{d_{pk}^2}{4\delta_s t}\right) + \text{Ei}\left(-\frac{d'^2_{pk}}{4\delta_s t}\right)\right]\right\} \\ \mathcal{S} =& \{1, 2, \dots p-1, p+1, \dots N_\varphi\}\end{aligned} \tag{3.38}$$

3.3.2 *Varying load Response of a Cable, and Convolution Model*

Equation (3.37) applies to the thermal transient in any underground cable subjected to a constant load step causing a constant power density applied to the cable conductor. The Fourier thermal conduction equation Equation (3.1) describes a Linear Time-Invariant (LTI) process. Therefore, the principle of superposition can be employed to obtain the time response to a time-varying load (G. J. Anders, 2005, Chapter 1). Defining a thermal impulse response to the system as $\theta_\delta(t)$, the response of the system to any varying load $W(t)$ can be described by the convolution

$$\theta(t) = \theta_\delta(t) * W(t) = \int_{-\infty}^{t} \theta_\delta(t-\tau)W(\tau)\mathrm{d}\tau = \int_{0}^{\infty} \theta_\delta(\tau)W(t-\tau)\mathrm{d}\tau \tag{3.39}$$

because $\theta_\delta(t) = 0$ for $t < 0$, due to the process being causal.

Additionally, the resistivity of the conductors varies with the temperature, and it increases the dissipated power for a given current load, as the cable conductor temperature rises. Obtaining a solution for the transient response of the cable under a varying resistivity can be challenging, and considering the resistivity at the steady state condition for calculating the transient temperature values can lead to temperature estimations that are higher than the actual temperatures.

(Goldenberg, 1967) provided upper and lower bounds for the temperature values in these circumstances. The upper bound formula was adopted by the standard (IEC, 2008)

$$\theta(t) < \frac{\theta_M(t)}{1 + \alpha_{\mathrm{c}}\left(\theta_M(\infty) - \theta_M(t)\right)} \tag{3.40}$$

where $\theta_M(t)$ is the maximum temperature rise calculated for time t, considering that the heat rate is that of the cable in steady-state. That is, calculated employing the resistivity at the steady-state conditions. $\theta_M(\infty)$ is the steady-state temperature rise, α_{c} is the temperature coefficient of the conductor electrical resistivity and $\rho_{\mathrm{c},\theta_{\mathrm{ref}}}$ is the electrical resistivity of the conductor at a reference temperature θ_{ref}, being $\rho_{\mathrm{c}}\left(\theta\left(t\right)\right) = \rho_{\mathrm{c},\theta_{\mathrm{ref}}}\left(1 + \alpha_{\mathrm{c}}\left(\theta(t) - \theta_{\mathrm{ref}}\right)\right)$.

In order to obtain the steady state temperature the following procedure can be followed. From Equation (A.12), for any κ constant $\mathrm{E}_1\left(\frac{\kappa}{t}\right)$ can be expressed as

$$\mathrm{E}_1\left(\frac{\kappa}{t}\right) = -\gamma - \ln\frac{\kappa}{t} - \sum_{j=1}^{\infty}\frac{(-\frac{\kappa}{t})^j}{j\,j!} \tag{3.41}$$

where γ is the Euler–Mascheroni constant. The difference between two such functions with κ_1 and κ_2 as constants as described by the mutual heating and soil effect processes,

$$\mathrm{E}_1\left(\frac{\kappa_1}{t}\right) - \mathrm{E}_1\left(\frac{\kappa_2}{t}\right) = -\ln\frac{\kappa_1}{t} + \ln\frac{\kappa_2}{t} - \sum_{j=1}^{\infty}\frac{(-\frac{\kappa_1}{t})^j + (\frac{\kappa_2}{t})^j}{j\,j!} \tag{3.42}$$

Therefore, in steady state,

$$\lim_{t\to\infty}\mathrm{E}_1\left(\frac{\kappa_1}{t}\right) - \mathrm{E}_1\left(\frac{\kappa_2}{t}\right) = \ln\frac{\kappa_2}{\kappa_1} \tag{3.43}$$

Applying Equation (3.43) to Equation (3.36) and Equation (3.35), steady-state temperature rise values for the mutual heating and soil effect processes can be calculated as

$$\theta_{pk}(\infty) = W_{\mathrm{tot}}\frac{\varrho_{\mathrm{s}}}{2\pi}\ln\frac{d'_{pk}}{d_{pk}} \tag{3.44}$$

$$\theta_e(\infty) = W_{\mathrm{tot}}\frac{\varrho_{\mathrm{s}}}{2\pi}\ln\frac{4L_d}{D_e} \tag{3.45}$$

3.3.3 *Electro-Thermal Degradation Models*

When XLPE cables are properly insulated from humidity electro-thermal stress is considered the main degradation factor (Mazzanti, 2009). Consequently, the degradation of the cable can be estimated, either directly through the measurement of the temperature, or indirectly, by estimating the cable temperature from the cable load (Han, H. M. Lee, and Shin, 2017). Multiple polymer degradation models exist that estimate the condition of the insulation of the cables, including Arrhenius, Zhurkov, and Crine models (Arrhenius, 1889; Zhurkov, 1984; J.-P. P. Crine, 2005; Aras, Alekperov, et al., 2007). A summary of this research can be found in (Montanari, Mazzanti, and Simoni, 2002).

Arrhenius formulated the relationship of chemical reaction rate with temperature and it was later adapted by (Dakin, 1948) as

$$\tau = \tau_0 \mathrm{e}^{\frac{u_{\mathrm{Arr}}}{k_B\vartheta}} \tag{3.46}$$

where U_{Arr} is the activation energy, k_B is the Boltzmann constant, and ϑ is the absolute temperature. Mazzanti combined the Arrhenius thermal degradation model with the Inverse Power Model (IPM) model for the electrical degradation, into a multi-stress electro-thermal degradation model of XLPE insulators as (Mazzanti and Montanari, 1997):

$$\tau = \tau_0 \mathrm{e}^{-\frac{U_{\mathrm{Arr}}}{k_B}\left(\frac{1}{\vartheta_0}-\frac{1}{\vartheta}\right)} \left(\frac{E}{E_0}\right)^{-\left(n_0 - b\left(\frac{1}{\vartheta_0}-\frac{1}{\vartheta}\right)\right)} \tag{3.47}$$

where ϑ_0 is a reference temperature such as room temperature, E_0 is an electrical field strength reference or threshold below of which the degradation is considered negligible, n_0 is the Voltage Endurance Coefficient (VEC) , b is a parameter that defines the synergism between the electrical and thermal stresses, and τ_0 is the lifetime at $\vartheta = \vartheta_0$ and $E = E_0$.

According to the Zhurkov model for lifetime of a solid subjected to a tensile stress σ can be described as

$$\tau = \tau_0 \mathrm{e}^{\frac{U_0 - \gamma\sigma}{k_B \vartheta}} \tag{3.48}$$

where k_B is the Boltzmann constant, ϑ is the absolute temperature, U_0 is the energy barrier that determines the probability that the bonds of the material are broken, and γ is a parameter related to the disorientation of its structure. This formulation was later adapted to represent the chemical degradation of cable polymers due to thermo-electrical stress by (Mamedov et al., 2004) as

$$\tau = \tau_0 \mathrm{e}^{\frac{w_0 - \chi E}{R_G \vartheta}} \tag{3.49}$$

where w_0 is the energy barrier or activation energy analogous to U_0, E is the electrical field strength applied, χ is a structural parameter of the polymer, and R_G is the universal gas constant.

The laboratory failure statistics can be adapted to a large size uniform cable polymer structure by employing the *enlargement law* for the cable enlargement in the radial direction (Hauschild and Mosch, 1992, Chapter 6).

$$\frac{E_{d63,1}}{E_{d63,2}} = \left(\frac{l_2}{l_1}\right)^{1/\beta_E} \left(\frac{r_{2,\mathrm{in}}}{r_{1,\mathrm{in}}}\right)^{2/\beta_E} \left[\frac{\left(\frac{r_{2,\mathrm{out}}}{r_{2,\mathrm{in}}}\right)^{2-\beta_E} - 1}{\left(\frac{r_{1,\mathrm{out}}}{r_{1,\mathrm{in}}}\right)^{2-\beta_E} - 1}\right]^{1/\beta_E} \tag{3.50}$$

where $E_{d63,x}$ is the 63% quantile of the Weibull distributed breakdown electric field strength for the cable x, and β_E is the shape parameter of the Weibull distribution.

(Mazzanti, 2009) used Equation (3.50) for scaling laboratory sample failure probabilities to a full sized cable defining a *enlargement factor* $\mathcal{D}$ as the factor where cable 1 are the laboratory sample cables (denoted with $\mathcal{S}$ subindex), and cable 2 is the full sized cable under analysis (denoted with $\mathcal{F}$ subindex):

$$\mathcal{D} = \left(\frac{E_{d63,\mathcal{S}}}{E_{d63,\mathcal{F}}}\right)^{\beta_E} = \left(\frac{l_{\mathcal{F}}}{l_{\mathcal{S}}}\right) \left(\frac{r_{\mathcal{F},\mathrm{in}}}{r_{\mathcal{S},\mathrm{in}}}\right)^2 \left[\frac{\left(\frac{r_{\mathcal{F},\mathrm{in}}}{r_{\mathcal{F},\mathrm{out}}}\right)^{\beta_E-2} - 1}{\left(\frac{r_{\mathcal{S},\mathrm{in}}}{r_{\mathcal{S},\mathrm{out}}}\right)^{\beta_E-2} - 1}\right] \tag{3.51}$$

The unavailability Q of a cable that follows a Weibull distribution, or probability P_F that the cable is faulted at a given time t, and therefore the Time To Failure (TTF) τ_F can be rewritten as (Meeker and Escobar, 1998, Chapter 4)

$$Q = P_F = \mathcal{P}\left(\tau_\mathrm{F} < t\right) = F(t) = 1 - \mathrm{e}^{-(\tau_\mathrm{F}/\lambda)^{\beta}} \text{ , or} \tag{3.52}$$

$$\tau_\mathrm{F} = \lambda \left(-\ln\left(1 - P_F\right)\right)^{1/\beta} \tag{3.53}$$

where λ is the scale parameter, and β is the shape parameter of the Weibull distribution. (Mazzanti, 2009) used the Zhurkov electro-thermal lifetime in Equation (3.49) as the scale parameter λ (or 63% probability to have failed before that time) in Equation (3.53), resulting

$$\tau_\mathrm{F} = \tau_0 \mathrm{e}^{\frac{w_0 - \chi E}{R_G \vartheta}} \left(-\ln\left(1 - P_F\right)\right)^{1/\beta} \tag{3.54}$$

and proceeding similarly for the Arrhenius-IPM model in Equation (3.47)

$$\tau_\mathrm{F} = \tau_0 \mathrm{e}^{-\frac{u_\mathrm{Arr}}{k_B}\left(\frac{1}{\vartheta_0} - \frac{1}{\vartheta}\right)} \left(\frac{E}{E_0}\right)^{-\left(n_0 - b\left(\frac{1}{\vartheta_0} - \frac{1}{\vartheta}\right)\right)} \left(-\ln\left(1 - P_F\right)\right)^{1/\beta} \tag{3.55}$$

Applying the enlargement law to the Weibull distribution (Mazzanti, 2009)

$$\frac{\tau_{\mathrm{F},\mathcal{F}}}{\tau_{\mathrm{F},\mathcal{S}}} = \left(\frac{\ln\left(1 - P_{F,\mathcal{F}}\right)}{\mathcal{D} \ln\left(1 - P_{F,\mathcal{S}}\right)}\right)^{1/\beta} \tag{3.56}$$

where $\tau_{\mathrm{F},\mathcal{F}}$ and $\tau_{\mathrm{F},\mathcal{S}}$ are the full cable and sample lifetimes at the respective design failure probabilities $P_{F,\mathcal{F}}$ and $P_{F,\mathcal{S}}$. Applying Equation (3.56) into Equation (3.55) and Equation (3.54)

$$\tau_{\mathrm{F},\mathcal{F}} = \tau_0 \mathrm{e}^{-\frac{u_\mathrm{Arr}}{k_B}\left(\frac{1}{\vartheta_0} - \frac{1}{\vartheta}\right)} \left(\frac{E}{E_0}\right)^{-\left(n_0 - b\left(\frac{1}{\vartheta_0} - \frac{1}{\vartheta}\right)\right)} \left(\frac{-\ln\left(1 - P_F, \mathcal{F}\right)}{\mathcal{D}}\right)^{1/\beta} \tag{3.57}$$

for the Arrhenius-IPM model and

$$\tau_{\mathrm{F},\mathcal{F}} = \tau_0 \mathrm{e}^{\frac{w_0 - \chi E}{R_G \vartheta}} \left(\frac{-\ln\left(1 - P_{F,\mathcal{F}}\right)}{\mathcal{D}}\right)^{1/\beta} \tag{3.58}$$

for the Zurkhov electro-thermal model

Mazzanti applied the fatigue theory proposed by Palmgren and Miner. This hypothesis is known not to always hold valid, but considered to be close for the maximum damage tests (Wilkins, 1956; Milne, Ritchie, and Karihaloo, 2003). The hypothesis states that the life reduction caused by a certain stress level is always the same, and the sum of life fraction losses up to the cable failure time equals to 1. That is,

$$\sum_{i=1}^{N_i} \Delta\xi_{i=1} \tag{3.59}$$

$$\Delta\xi_i = \frac{\Delta t}{\tau(\sigma_i)} \tag{3.60}$$

where N_i is the number of time intervals up to the failure time, $\Delta\xi_i$ is the fraction of life loss or degradation increment caused by each i^{th} stress level σ_i, during a Δt interval.

Therefore, for a continuously varying cable temperature and electric field, the cable degradation ξ at any time instant can be estimated as (Mazzanti, 2009)

$$\xi(t) = \xi(K\Delta t) = \sum_{i=1}^{K} \Delta\xi_i = \sum_{i=1}^{K} \left\{ \int_{(i-1)\Delta t}^{i\Delta t} \frac{\mathrm{d}t}{\tau(\sigma(t))} \right\} \tag{3.61}$$

$$= \sum_{i=1}^{K} \left\{ \int_{(i-1)\Delta t}^{i\Delta t} \frac{\mathrm{d}t}{\tau_{\mathrm{F},\mathcal{F}(E(t),\vartheta(t))}} \right\} \approx \sum_{i=1}^{K} \left\{ \frac{\Delta t}{\tau_{\mathrm{F},\mathcal{F}(E(t),\vartheta(t))}} \right\} \tag{3.62}$$

where K is the number of Δt intervals up to the estimation time t. The approximation can be considered accurate as far as the interval Δt is small with respect to the dynamics of the electric field and temperature.

Figure 3.13 shows the degradation of cable Cable A in Appendix A.1 for each hour at the given electrical field and temperature values, according to the Zhurkov based model.

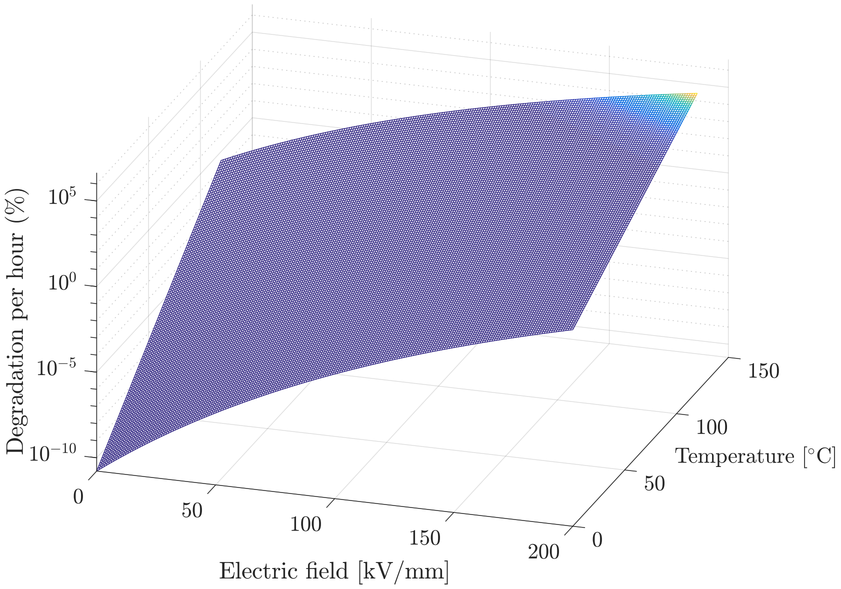


Figure 3.13: Degradation of XLPE cable insulator vs temperature and electrical field.

3.3.4 *Factors that affect the thermal model estimations*

The thermal model simulations are shown to be accurate as far as all the parameters are known with accuracy. However, the results are affected by several parameters such as the cable dimensions, installation, ambient temperature and humidity. The latter two vary with time, and they can be included as additional sources of uncertainty in the system model. One of the main factors affecting the thermal model is considered to be the thermal resistivity of the soil (Alwan et al., 2016; Diaz-Aguilo and Leon, 2015).

It has been shown that the thermal diffusivity of the soil depends on its moisture content, and the value can be estimated as (G. Anders and El-Kady, 1992 and IEC, 2008, Annex D)

$$\delta_{\mathrm{s}} = \frac{1 \times 10^{-3}}{\varrho_{\mathrm{s}} d_{\mathrm{s}} (0.82 + 0.042\phi_{\mathrm{s}})} \tag{3.63}$$

where ϱ_s is the thermal resistivity of the soil, d_s is the dry density of the soil and ϕ_s is the moisture content of the soil in % of the dry weight. When other parameters are unknown, a typical accepted value for the diffusivity is $0.5 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$, corresponding to a soil resistivity of $1\,\mathrm{K\,m\,W^{-1}}$ and a 7 % moisture content. However, the thermal resistivity can generally vary in a range from 0.5 to $3.0\,\mathrm{K\,m\,W^{-1}}$ and the thermal diffusivity from 0.2×10^{-6} to $0.8 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$ (IEC, 2008, Annex D).

4

MC SIMULATION OF SAN WITH PHYSICS-BASED MODEL ACTIVITES, BASED ON FPGA

One of the objectives of this thesis is to produce a parallel simulation method for Stochastic Activity Network (SAN) models that is capable of performing the Monte Carlo (MC) simulation of complex SAN models that include physics-based component models in addition to stochastic models (see Section 1.1).

The MC simulation of a SAN model is inherently different from the simulation of Fault Trees (FTs). The simulation of FT involves the generation of the input events according to their statistical properties, but independently from the FT structure, unless model-dependent optimizations are used for its simulation (Ruijters and Stoelinga, 2015). When basic events are independent from each other, they can be generated in parallel, and later introduced to the system, in order to evaluate the system failure state. This method permits the accelerated simulation of a FT in Field Programmable Gate Array (FPGA), by parallelization of the generation of input basic event sequences and a parallel analysis of the FT result, employing combinational or sequential propagating circuits (Aliee and Zarandi, 2013).

In comparison to FT events, events in a SAN model are governed by the activities in the SAN structure, and the events cannot be generated in parallel, given that activities are not independent of each other. The completion of some activities cause other activities to become enabled and their events to be triggered according to a probability distribution that depends on the newly generated marking.

Therefore, the events in each trial in the MC simulation of a SAN model follow a random sequential process (Nelson, 2013, Chapter 3). No event has a pre-established completion time and the activation and completion of a given activity ac_i depends on the completion of other activities (see Section 2.2).

However, at each activity completion, the new state must be evaluated for all input gates and activities in the system. This can be a slow process to perform sequentially by a Central Processing Unit (CPU) for larger SAN models.

Additionally, due to the rarity of failure events in high-reliability systems, a large number of MC trials must be performed for obtaining accurate results (see Chapter 2), and the simulation of each MC trial with the addition of physical models in a CPU can be challenging.

Some authors have approached the failure probability analysis of equipment based on FPGA (Shoar et al., 2017; Ejlali and Miremadi, 2004; Aliee and Zarandi, 2013; Aliee and Zarandi, 2011). However, the methods are based on formalisms where failure events are caused by a certain ordering or sequencing of the input events, or by the events being generated in certain time intervals. These mechanisms do not suffice to represent the complex interdependency between activities that causes the event sequencing in a SAN model and the component degradation models involved in the simulation of an electric power distribution grid.

In this context, this research concentrates on employing FPGA for an accelerated simulation of SAN model. The results of this simulation permit obtaining the reliability estimates of power cables. This section discusses and describes methods for the simulation of such SAN models within an FPGA and compares the results to CPU and Graphics Processing Unit (GPU) based solutions.

4.1 Monte Carlo Timing, Data Collection and Results Analysis

As discussed in Section 2.6.1, the simulation of the events of in an electric power distribution grid requires the generation of very rarely occurring events with long time-intervals the simulation of long mission times. At the same time, the simulation of cascading failures and operational switching events requires that short time intervals are also accurately represented, forcing a higher MC time resolution.

In this context, it is possible to perform the full MC simulation by simulating each component in continuous time and capturing their state values in larger discrete-time intervals. However, certain components such as circuit breakers may operate every few hours which may change the state of the system, and they can affect the degradation process of other components (see Section 2.6.1).

Therefore, the sampling of the simulation needs to be performed considering the fastest operational intervals and the degradation process dynamics. However, performing this sampling based on fixed short intervals over the period of several decades that takes before any failure event arises is inefficient. This process is shown in Figure 4.1a. In this example, the grid is subjected to short-interval operational events at a rate of λ_{op} and rare failure events at a rate of λ_F, that cause system state changes. When a fixed short time-interval is established, the simulation must be monitored at every interval, causing unnecessary state captures without any state change, that cause module synchronization pauses, state analysis and memory transfers.

An alternative is to only sample the system state in response to events that may change the model state as shown in Figure 4.1b.

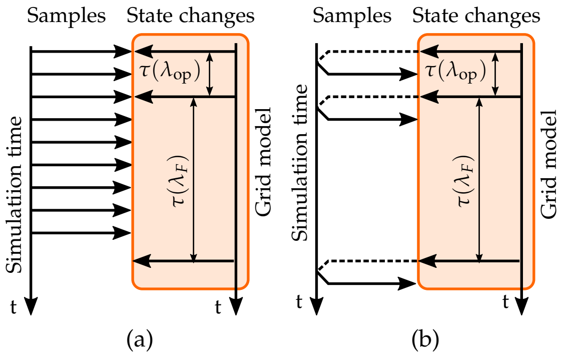


Figure 4.1: MC sampling comparison: (a) fixed interval sampling in comparison to (b) sampling in response to state changes.

The responsive sampling model shown in Figure 4.1b has the advantage of being an efficient process to capture the simulation state. Notice, however, that an event may not cause any state change in a system as shown in Figure 4.2.

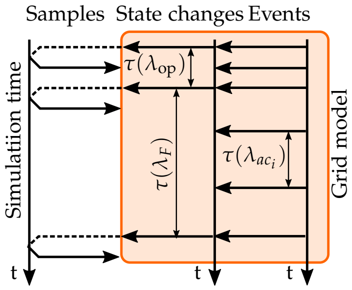


Figure 4.2: Events such as the ac_j activity completion may not cause any state changes. However, the state must be monitored after every activity completion.

In this example, after the completion of an activity ac_j, the system state remains in the same state. The state must still be observed to detect any state changes or events/conditions that may be desirable to be captured.

Thus, the simulation of a SAN model that describes the system must be continuously monitored for any state change at every activity completion, even if the completion causes no marking changes in the places.

Therefore, the simulation method adopted in this thesis is based on a SAN model simulation that drives the capture process of the MC simulation. As a result, the sample time intervals of the MC simulation will be non-uniformly distributed. The main advantage of this method is the reduced number of state samples required to be captured for obtaining the MC simulation results, and the reduced memory resources to record them. Given that the time resolution is fixed by the model itself, this method also prevents issues with the sequencing errors and uncaught events that could be caused by the sample time interval or the time value resolution.

4.2 CPU and GPU in comparison to FPGA Implementations

A SAN model simulation including physical component models involves the execution of 3 processes:

1. Simulation of the SAN grid;
2. Simulation of the physical degradation models within certain activities (see Section 2.3);
3. Capture of the results when states (marking of the SAN places) change.

Those three processes can be executed in parallel, but not independent of each other. That is, the simulation of the power grid controls the component degradation models, and the component degradation models control the behavior of the grid (see Section 2.3). Therefore, a sequence must exist for the simulation of the different processes, and the event/activity intervals that govern the model cannot be independently generated nor simulated. Additionally, the capture process of the events should be performed by request of the activity completion events (responsive capture process) in order to reduce resources and simulation time, as described in Section 4.1.

However, the SAN model of a full grid similar the one shown in Figure 2.15 can be large and involves the execution of hundreds of activities. The sequential processing of the activities in either CPU is impractical, and their acceleration employing Vector Floating-Point Unit (VFPU) is not optimal, given that activities are not uniform and do not operate with the same parameters. When certain activities involve physical degradation models, further complex algorithms need to be performed in-CPU, which limits the speed of the simulation of the rest of activities.

Current embedded systems, typically offer the structure shown in Figure 4.3, where the multiple processes in the SAN simulation can be run and parallelized either on a multi-core CPU (including their VFPUs), on a GPU, in an FPGA Programmable Logic (PL) or in an Application-Specific Integrated Circuit (ASIC). The results can be recorded and manipulated in FPGA Block RAM (BRAM), external Random Access Memory (RAM), or On-Chip Memory (OCM). However, the latencies involved in the communication buses between the separate components such as the multiple cores in a GPU or between CPU and GPU or memory can limit the performance of the simulations (Papakonstantinou et al., 2009).

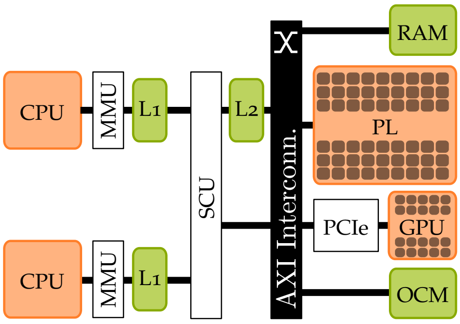


Figure 4.3: Typical interconnection architecture of CPU, GPU and PL.

When parallelization in both CPU and PL is considered, even with Advanced Extensible Interface (AXI) buses clocked at 333 MHz found on certain current FPGA devices (Xilinx Inc., 2018b) and employing 128 bit bus widths, the sequential read-writes to the hundreds of components can drop the performance considerably. Therefore, it is desirable that the simulation is run standalone within a single component (CPU, GPU or PL) and results transferred only when after the simulation or part of it is executed.

The execution of generic Petri net structures in GPU has also been proven to be complex, and their efficiency is only proven to be valid when the Software (SW) is tailored to run specific Petri nets that are adapted for simulation (Yianni et al., 2018). When heterogeneous grids and models are under consideration, this task can be challenging.

On the other hand, FPGA architectures are ubiquitous to represent structures such as that of a SAN model. FPGAs inherently include mechanisms that interconnect components with low latency buses. Formalisms such as Petri Net (PN) have been executed in FPGA (Soto and Pereira, 2005; Leroux, Andreu, and Godary-Dejean, 2015; Chang, Kwon, and Park, 1996; Bukowiec and Adamski, 2012) and they have also been proven to be efficient for signal processing and algorithm acceleration, which can be employed to accelerate the simulation of degradation mechanisms.

Therefore, one of the goals of the present thesis is the execution of a full grid model based on the SAN formalism, entirely in the PL part of the FPGA, which must also be capable to include physical degradation models following the framework described in Chapter 4.

4.3 High Level Synthesis vs Hardware Description Language Implementations

When developing FPGA architectures four approaches can be taken:

1. The development of every component in the PL using low-level Hardware Description Language (HDL), such as VHSIC Hardware Description Language (VHDL) or Verilog.
2. The development of the design using High-Level Synthesis (HLS), such as Vivado HLS (Xilinx Inc., 2018a) or PandA Bambu (*PandA Bambu* 2019).
3. The development of the design employing block diagrams.
4. A hybrid approach using a combination of the previous techniques for each component and system level designs.

The traditional method of employing *low-level* programming languages requires detailed knowledge of the inner architecture and it can be error prone. Therefore, HLS methods have been developed by multiple manufacturers, including Vivado HLS (Xilinx

Inc., 2018a), PandA Bambu (*PandA Bambu* 2019), Catapult HLS (Mentor Graphics, 2019) or SystemC (IEEE, 2012). HLS methods mostly rely on sequential or object-oriented programming, generally C or C++, in a similar manner to the programming of a CPU, or model-driven development with automated code generation. The code is then parsed by a SW and patterns are identified for components that can be parallelized or accelerated in hardware. However, identifying the parts that can be accelerated often requires manual annotation (Cong et al., 2011), and current existing HLS solutions may require manual modifications and optimization of the code so that they can be synthesized and often result in architectures that are less efficient than traditional SW solutions (Fibich et al., 2018; Cong et al., 2011). Compared to this approach, HDL languages permit tuning each single clock cycle in the system.

The block diagram approach is offered by some manufacturers, as an aid for quick prototyping. The advantage of the method is an instantaneous visualization of the developed architecture, which can be defined from existing design diagrams. Each block in the diagram can accomodate designs developed in either HLS, HDL or block diagram methods, producing a hybrid language result. Additionally, manufacturers generally offer pre-defined Intellectual Property (IP) blocks that can be easily employed in these designs for producing efficient solutions. They permit the development of optimized architectures by researchers on other fields with limited knowledge about HDL languages.

Block diagrams permit the translation from existing block designs produced in SW from other research fields, and the validation of the designs through visual inspection by field experts. However, the block diagram based design method is frequently limited by the capability of the user interface.

The validation of the methods described throughout the thesis have been performed by implementations done with hybrid designs employing both VHDL languages and block diagrams. However, no assumption has been made about the development methods used for the development of the architecture. The methods described can be applied to any PL development flow, as far as the tools employed permit the flexibility to produce and synthesize the described architectures.

4.4 PSEUDO-RANDOM NUMBER GENERATORS

In order to generate the stochastic time intervals that govern SAN activities, it is necessary to generate random numbers that are distributed as described by each activity Probability Density Function (PDF). It is usual that these random numbers are generated by Pseudo-Random Number Generation (PRNG) algorithms that generate sequences that have long periods before they restart the initial sequence (Matsumoto and Nishimura, 1998). It is necessary that the periods are long enough to be able to produce random numbers for the full MC simulation (Srinivasan, Mascagni, and Ceperley, 2003).

The generation of pseudo-random numbers for simulating the stochastic process events is critical because it could compromise the validity of the results (Zio, 2013, Chapter 3). For modeling high-reliability components, these PDFs must represent rare events that require a large number of numbers to be generated for an accurate representation (see Section 2.6 and (Rubino and Tuffin, 2009)). Therefore, the employed PRNG must be able to generate a large number of independent random numbers, which need to be uncorrelated for each stream that feeds a component and also in between independent components. These characteristics must be validated by applying statistical tests on the generated random numbers (Srinivasan, Mascagni, and Ceperley, 2003).

Methods for the generation of randomly distributed numbers often rely on Uniform Random Number (URN) PRNG (Zio, 2013, Chapter 3,Kroese, Taimre, and Botev, 2011, Chapter 3). Commonly, the methods to sample for any random PDF can be grouped into three sets (Thomas, Luk, et al., 2007)

1. Transformation methods.
2. Inverse Cumulative Distribution Function (ICDF) based methods.
3. The Von Neumann algorithm or sample rejection method.

Transformation methods are employed when specific functions can be applied to a certain distribution in order to obtain samples of another distribution, such as in the Box-Muller transformation (Danger et al., 2000) or the Wallace method (D. U. Lee et al., 2005). However, their application is limited to distributions where such properties are available. The following sections describe the other two methods that are more generalistic.

4.4.1 *ICDF methods*

Let X_U be a uniformly distributed set of N random numbers

$$X_U \sim \mathrm{U}[0,1) \tag{4.1}$$

Being $f_\iota(\tau)$ any PDF function limited between (a,b) that describes the time interval sequence $\iota = \{\tau_0, \tau_1, \dots\}$ for an activity in SAN, or simply random numbers required for a MC simulation, its Cumulative Distribution Function (CDF) F_ι is defined as (Zio, 2013, Chapter 3)

$$F_\iota(\tau) = \mathcal{P}\left(\iota \leq \tau\right) = \int_{-\infty}^{\tau} f_\iota(t)dt \tag{4.2}$$

and its inverse function for any $y \in [0,1)$ as

$$F_\iota^{-1}(y) = \inf\{\tau : F_\iota(\tau) \geq y\} \tag{4.3}$$

Defining a new sequence Y that is obtained by applying the inverse cumulative distribution function F_ι^{-1} to the given uniformly distributed number sequence X_U, the new sequence obtained is distributed according to f_ι, given that (Zio, 2013, Chapter 3):

$$\begin{aligned} \iota &= F_\iota^{-1}(X_U) && (4.4)\\ \mathcal{P}\left(\iota \leq \tau\right) &= \mathcal{P}\left(F_\iota^{-1}(X_U) \leq \tau\right) && (4.5)\\ &= \mathcal{P}\left(X_U \leq F_\iota(\tau)\right) && (4.6)\\ &= F_\iota(\tau) && (4.7) \end{aligned}$$

and therefore the newly generated set of numbers ι is distributed $\iota \sim f_\iota(\tau)$.

This method requires that a ICDF exists for the given PDF. However, in terms of Hardware (HW) implementations it may require floating-point operations using large resources or complex fitting algorithms to be performed, in order to evaluate the values of this ICDF (Gutierrez, Torres, and Valls, 2012; Schryver et al., 2010; McCollum et al., 2003; Thomas and Luk, 2006; Wei, Chengshu, and Xin, 2004).

4.4.2 *Von Neumann algorithm or Sample rejection method*

This method is adequate for complex PDF where the ICDF may not be available. This method is based on generating a uniformly distributed stochastic number sequence $X_U \sim U(a, b)$, where a and b are the limits of the desired PDF $f_\iota(\tau)$ and accepting or rejecting the samples according to $f_\iota(t)$, as follows (Zio, 2013, Chapter 3).

Let $h(\tau)$ be a normalized function of $f_\iota(\tau)$

$$h(\tau) = \frac{f_\iota(\tau)}{\max\limits_{\tau} f_\iota(\tau)} \tag{4.8}$$

A second uniformly distributed number sequence is generated $X'_U \sim [0, 1)$

A given number $x_i \in X_U$ is accepted or rejected according to the second generated random number $x'_i \in X'_U$. x_i is

1. Accepted if $h(x_i) \geq x'_i$
2. Rejected otherwise.

Figure 4.4 shows graphically the generation of the random numbers and Figure 4.5 shows a block diagram for the algorithm. Notice that the block diagram is not implementable as it does not consider block delays or signal synchronization as it will be discussed in the following section.

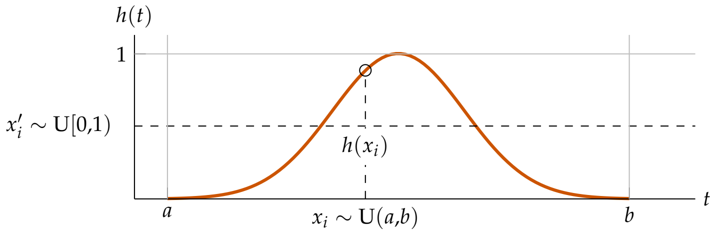


Figure 4.4: Rejection method for arbitrary PDF sample generation

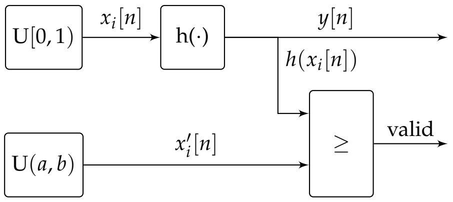


Figure 4.5: Rejection method block diagram (block delays not considered).

The main problem of this method is that a large number of samples are rejected, especially on low probability value samples. Therefore, the algorithm can be very inefficient.

4.4.3 *FPGA based PRNG*

Both the ICDF and the rejection methods require a uniform distribution PRNG. The use of Linear Feedback Shift Registers (LFSRs) is the most common method to generate URN pseudo-random numbers both in SW and also in FPGA due to their relatively low complexity and high efficiency and for their capability to produce large period sequences (Kroese, Taimre, and Botev, 2011; Ortega-Zamorano et al., 2016). PRNGs employed for statistical analysis must exhibit good statistical properties with no recognizable patterns, which is assured by passing multiple tests (Srinivasan, Mascagni, and Ceperley, 2003) such as Knuth tests (Taussky and Knuth, 2006), Diehard Battery Tests of Randomness (Marsaglia, 2003) or the Kolmogorov-Smirnov test (Pecht, 2008, Section 3.3). It is also required that it has a long periodicity, in order to satisfy the large number sets required for the MC simulations of high reliability equipment. The URN generators are in continuous evolution and their statistical properties are still being discussed. *Mersenne Twister* is a very common algorithm with long periodicity (Kumamoto and Henley, 1996). Xorshift algorithms are a particularly efficient subset of LFSR discovered by (Marsaglia, 2003) but had originally failed several statistical tests. Xoroshiro128+ is an evolution of the xorshift algorithms of Marsaglia, that improves on their statistical properties, and requires very low resources to be implemented in FPGA (Blackman and Vigna, 2018). The xoroshiro128+ algorithm has been employed throughout this thesis due to its long 2^{128} period that fulfilled the requirements of periodicity for the simulations in this thesis. However, larger periods and improved statistical properties can be achieved for more complex models (Blackman and Vigna, 2018).

Figures 4.6 and 4.7 describe the block diagrams of the ICDF and Von Neumann methods as they would be implemented in an FPGA. In these diagrams n is the discrete time instant index. The ICDF architecture can employ either floating-point operators or a Lookup Table (LUT) based polynomial approximations (Gutierrez, Torres, and Valls, 2012) for an accurate representation of the normalized PDF $h(\tau)$ (Gautham, 2010). There are also specific methods for accelerating the ICDF (Schryver et al., 2010).

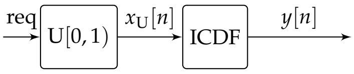


Figure 4.6: ICDF method.

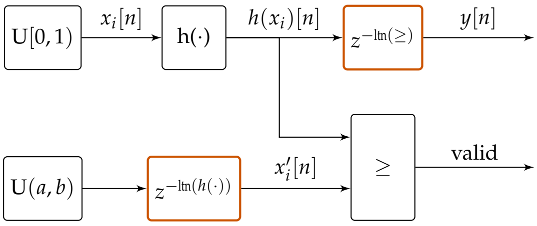


Figure 4.7: Rejection method algorithm implementation with added data alignment delay blocks.

Even when floating-point units are used for the ICDF, if a pipelined architecture is utilized for the operators, the architecture can be capable of producing a new sample for each clock. Therefore, ICDF methods can require large resources but offer a very fast throughput. If an expression for the inverse function of the CDF does not exist, it can be numerically approximated.

The main advantage of rejection based method is that they permit obtaining a very accurate approximation to the actual PDF (Thomas, Luk, et al., 2007). However, as it was described in Section 4.4.2, it can cause a large number of samples to be rejected, causing large delays and reducing the block throughput. Figure 4.7 describes the structure of a generic PRNG based on the rejection method. In an FPGA implementation, the calculation of the normalized PDF $h(\tau)$ can be pipelined. With this method, a pipelined structure permits obtaining a throughput of one sample per clock, generally at the cost of an increased total latency. However, the block can require multiple clock cycles to evaluate $h(x_i[n])$, while the second URN generator can typically deliver a sample in each clock. Therefore, data sample alignment is required by adding delay blocks. These delay blocks are typically implemented by shift-register blocks that produce a $z^{-\mathsf{ltn}(bk_i)}$ delay, where $\mathsf{ltn}(bk_i)$ is the block latency in clock cycles of the corresponding block bk_i. A pipeline architecture as shown in Figure 4.7 can produce a random number in its output at every clock cycle. However, a large fraction of the generated numbers are discarded later for not being valid, rendering the method less efficient than the ICDF methods.

Some methods such as the Ziggurat algorithm improve the efficiency of the rejection algorithms by increasing the acceptance ratio of the generated numbers by approximating $h(\tau)$ with rectangular areas and residual more complex densities (Marsaglia, Tsang, et al., 2000). The algorithm has also been adapted for HW implementations (G. Zhang et al., 2005).

4.4.4 *Generalized Random Number Generator for SAN*

The simulation of SAN model requires the simultaneous generation of several random numbers for each of the activities that are enabled at each simulation time step. SW simulations require that the PRNG of each activity is run sequentially, unless multiple PRNG are parallelized in the multiple available cores. In FPGA simulations, a separate PRNG can be assigned to each of the activities. Multiple PRNG can be combined into a single generator that can be embedded in each of the activities (Wei, Chengshu, and Xin, 2004). However, the use of resources would grow linearly with the size of the SAN network, which would have big impact on the resource usage of the simulation architecture.

The method would be especially inefficient, considering that only very few activities change their state at each time instant, and the parallelization of the PRNG requires ensuring that the samples from the different random sequences are not correlated. That is, ensuring that there is no *inter-stream* correlation which can be difficult to prove (Srinivasan, Mascagni, and Ceperley, 2003).

An alternative solution is the centralization of the PRNG for all the activities. Generally multiple activities in a SAN model follow the same probability distribution types (exponential, Gaussian, Weibull...) with different parameters (scale factor, rate, variance...). In many cases, the same PRNG block can be employed, samples can be adapted for the parameters of each activity and then distributed. The method ensures the samples from different activities to be uncorrelated, as far as the samples within the PRNG sequence are not correlated (*intra-stream correlation*) (Srinivasan, Mascagni, and Ceperley, 2003).

The adopted alternative is described for the case of a SAN model with four enabled activities in Figure 4.8. In this example, the intervals of two activities ac_1 and ac_2 are distributed according to the PDFs f_{ac_1} and f_{ac_2} which are of class f_1, while ac_3 and ac_4 are distributed according to the PDFs f_{ac_3} and f_{ac_4} which are of class f_2. The PRNG includes two generators, one for each of the probability distribution classes, that work in parallel. The module keeps buffers (registers) for a sample in each distribution generator block and activities are fed their intervals, immediately after a request (`req`) is received, if available. Whenever the interval is assigned the buffered value, a new value is immediately generated and stored in the buffer.

In this architecture the PRNG block works in parallel with the activity completions (the simulation of the SAN model). If a large enough buffer is provided to feed the minimum number of activities completing at each time instant, the activity requests can be satisfied in a single clock cycle.

4.5 SAN Models in FPGA

The execution of SAN models follows a sequence in which the activity completion is coordinated. That is, activities cannot be activated or completed independently without await-

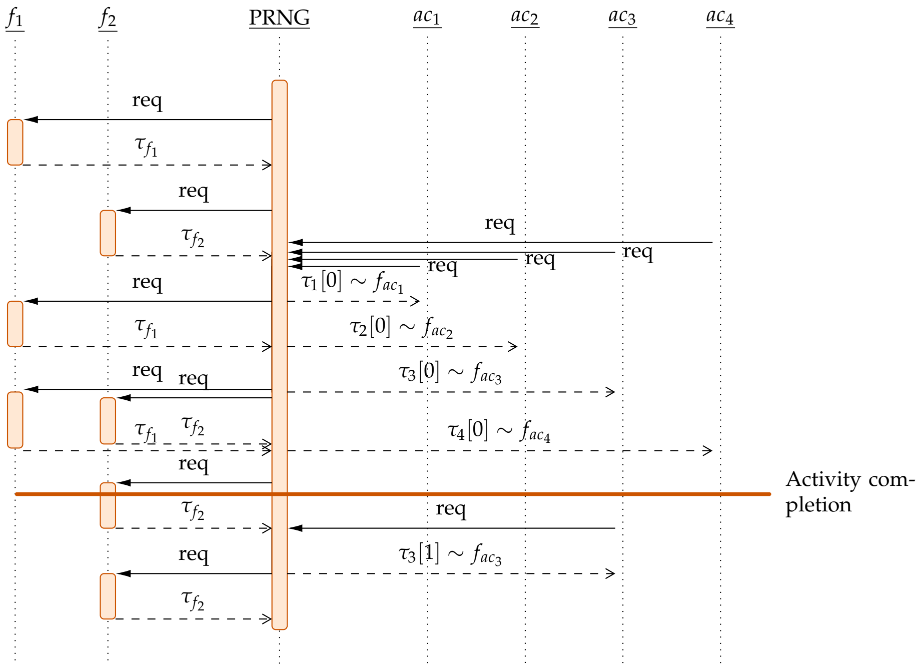


Figure 4.8: Sequence diagram of the centralized PRNG.

ing the (stochastic) actions in other activities. Theoretically, any activity could act on its own if it had a time reference at any instant. However, in that case the simulator would need to go through every single time step, because the activities may switch their states at any given time instant.

As an example, lets assume that a given cable has an expected lifetime of 50 years and the minimum sampling interval for certain component is 5 min. If the simulation is run on an FPGA at 300 MHz clock frequency, each simulation would require 17.5 ms just to generate the random time intervals. On a MC simulation with 1×10^6 trials, the method would require 4.9 h solely on counting the time. As described in Section 4.1, it is more efficient to evaluate the model only at every time instant when an activity completion event occurs. Due to the low frequency of occurrence of rare events in high reliability equipment models, the described simulation would involve very few time counts. For example, an electric power distribution grid with 100 events would require 0.3 µs on time interval generation, resulting on a total of 3 s delay for the full MC simulation. Notice, however, that the simulation time is also limited by the sampling interval requirement for the physical models that may require frequent sampling to obtain results as it will be discussed in the following sections.

However, with this responsive to events based MC simulation method, activities generate time intervals in varying clock rates, and the time increase after each activity completion varies with the time-intervals generated by the activities. Therefore, there is no notion of a shared simulation time-base for all activity blocks. Thus, activity blocks cannot count the time on their own, and the notion of a common shared time base must be controlled independently for the activity sequencing. Accordingly, a centralized scheduling architecture is required for the simulation management. This architecture also benefits from the centralized PRNG, as the random numbers do not need to be distributed to the activities, but only to this scheduling manager. This structure prevents the distribution of signals to hundreds of SAN activities, which could cause issues with latencies due to complex routing.

The proposed architecture therefore consists of three main elements:

- A scheduler block that governs the simulation time, activity sequencing and the generation of random samples for the activities.
- Activity blocks that govern the marking and execution of functions of the SAN model
- A PRNG block that generates the necessary distribution samples for the scheduler block.

The architecture is shown in Figure 4.9.

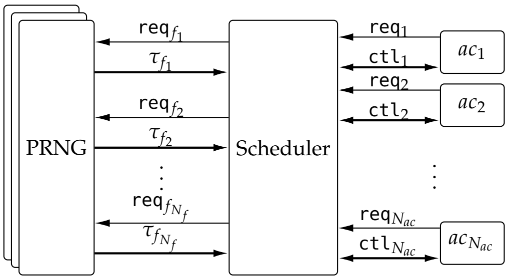


Figure 4.9: Simulation architecture for a centralized PRNG.

A PRNG block, capable of producing random numbers distributed according to N_f distributions $f_1 \dots f_{N_f}$ is connected to a scheduler that requests the generation of new samples and buffers the generated values.

The scheduler assigns time intervals to the activities at every request on signals $\mathtt{req}_i$ for each activity ac_i. Time intervals are generated according to the PDF parameters requested by each activity, which are transferred over the $\mathtt{ctl}_i$ control buses, that are used to read the activitiy configuration and state, and to govern their completion.

4.5.1 *Parallel SAN simulation HW architecture*

The architecture shown in Figure 4.9 is inherently parallel, because all activity blocks can evaluate their condition and complete the activities in parallel. However, due to the required sequentiation between activities, a single activity can complete at any given time instant. Given the architecture shown in Figure 4.9, the topology for an FPGA SAN simulator can be constructed as in Figure 4.10.

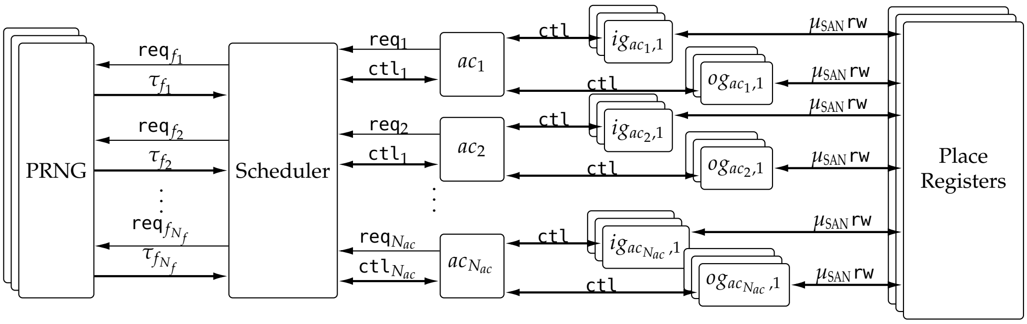


Figure 4.10: Topology for a distributed component FPGA SAN simulator.

The complexity of an efficient parallel simulation of a SAN model lies on the following factors:

- The enablement state changes of each of the activities must be monitored at every simulation time step;

- Pseudo-random numbers must be transformed and assigned to each of the activities at each request;
- The time intervals need to be sorted to obtain the next activity to complete;
- At each activity completion, multiple functions must be evaluated as defined by the input and output gates associated to the activity.
- The evaluation of the stochastic time intervals for activities related to physical models require the full (conditioned) simulation of the physical models.

The following sections Section 4.5.1.1 to Section 4.5.1.8 cover the architecture that addresses the above mentioned issues in order to accelerate the SAN simulation.

4.5.1.1 *Non-parallel HW simulation of a SAN model*

Sanders already described the simulation of flat SAN models and hierarchical SAN models in a single CPU (Sanders and Meyer, 1991a). The hierarchization is not an issue in an FPGA simulation because the modularization is part of the design model employed for hardware. HW simulations mainly benefit from two factors:

1. Granular parallelization of processes with smaller function blocks is possible thanks to the virtually unlimited parallelization in hardware.
2. Transfer of data to RAM memory is unnecessary because the data is held locally into registers.

However, due to the distribution of modules, the inter-module communication and sequentiation must be addressed. Block and process latencies can be variable in each model, e.g. more or less complex output gate functions with varying latencies. Therefore, the latencies must be controlled employing either time-alignment blocks or bidirectional *acknowledgement* (`ack`) signals. The architecture cannot be fully parallelized, as the sequentiation of events of the simulation is required, and some of the parallelization is conditioned to the structure of the model. Figure 4.11 shows a simplified flowchart diagram to be implemented by the scheduler block for the control of the architecture described in Figure 4.10.

The scheduler block must initially reset all the blocks to their initial state (including activities, PRNG, input and output gates and markings) in Process ①. The reset signal can be propagated hierarchically from the scheduler to the activities, down to the gates and places, simplifying this process, but latencies must be tightly controlled. Once all the blocks are reset, the scheduler waits for all activities to evaluate their enablement state (Process ②). At this point, activities evaluate the signals from the input gates that control their enablement. Once activity blocks have finished monitoring the enablement state, they generate requests for new pseudo-random intervals, and the scheduler evaluates the activities iteratively. If the activity requested the generation of a new random number (Condition ⓑ), a new pseudo-random number is collected (Condition ⓒ) and transformed to adapt to the parameters of the activity (Process ⑦). A previously active activity can request its cancellation if the enablement condition is no longer met (Condition ⓓ), and its completion time is disabled in the completion time list Ct (Process ⑥). Once all activities have been parsed (Condition ⓐ), the next activity to be completed ac_{comp} is calculated, by calculating the minimum completion time of the activities in the SAN model (Process ④), the activity is completed (Process ⑨). The scheduler then waits for the markings to be updated accordingly (Process ⑩). Once the markings have propagated, the new SAN state metrics (*rewards*) are captured and recorded for later evaluating the probability distributions of those rewards (Process ⑪).

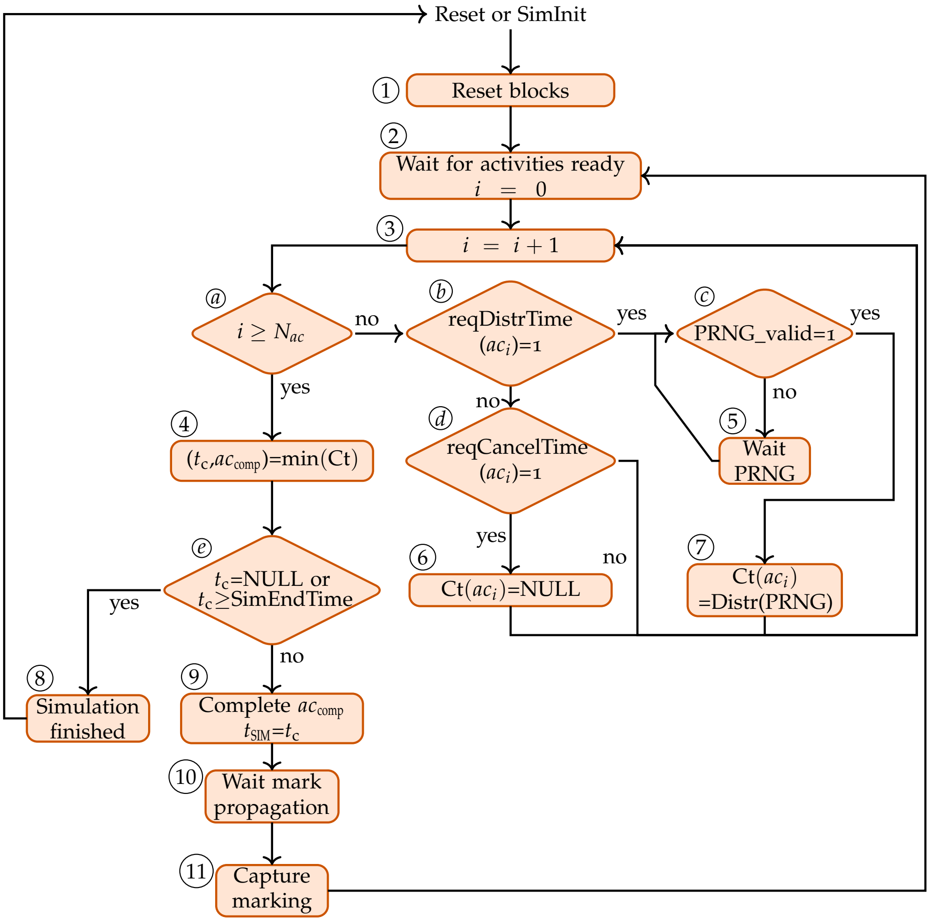


Figure 4.11: Flowchart diagram for a sequential SAN simulation.

In this flowchart, we can observe that:

- Part of the processes described in this flowchart consider the parallel execution in other blocks of the architecture shown in Figure 4.10. Activities, for example, can evaluate their enablement status independently. However, they cannot evaluate the condition at any time instant, because the place markings could have incorrect transient values;
- The collection of the enablement state of each of the activities is sequential, and the latency of this process grows linearly with the number of activities in the SAN model;
- The calculation of the next activity to be completed in Process ④ is performed by a sequential search algorithm, and the process latency grows linearly with the model size (measured by the number of activities N_{ac}), unless timings are sorted while generated;
- The completion of activities in Process ⑨ is a multi-cycle process that depends on the function and implementations of the input and output gates;
- The generation of random numbers in Process ⑦ can take advantage of the architecture shown in Figure 4.10.

4.5.1.2 *Parallel evaluation of the states of the activities*

As discussed in the previous section, activity blocks can be implemented as separate modules that process their instantaneous state in parallel to other processes. However, their enablement state cannot be evaluated unless the marking of the corresponding places is not updated after the previous activity completion. This situation happens because of the latency of the activity completion, and the latencies of the predicates and functions of the input and output gates. Those latencies vary depending on the complexity of the functions. Therefore, signaling needs to be employed to notify the functions have finished updating the markings.

Additionally, multiple activities can be enabled after the completion of each activity, but only a single activity can be completed at each stage (see Section 2.2). Figure 4.12 shows a minimal example of a SAN model with a single activity.

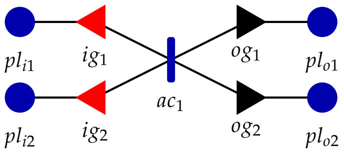


Figure 4.12: Minimal SAN model example with a single activity.

The activity (ac_1):

1. Evaluates the enablement by the set of input gates connected to it $\text{IG}(ac_1) = \{ig_1, ig_2\}$.
2. If the activity is enabled, it requests the scheduler the generation a new random interval $\tau_{ac_1} \sim f_{ac_1}$, distributed according to the associated PDF f_{ac_1}.
3. The activity awaits for the completion request from the scheduler. An activity may not complete if other activities complete earlier than the time instant when this interval τ_{ac_1} expires. Therefore, the activity block cannot complete on its own, independently and in parallel, and it is governed by the scheduler.
4. If the scheduler requests the completion of the activity, the activity block will:
    4.1. Request the execution of the functions of the input gates $f_g(ig_1)$ and $f_g(ig_2)$.
    4.2. Request the execution of the functions of the output gates $\text{OG}(ac_1) = \{og_1, og_2\}$, $f_g(og_1)$ and $f_g(og_2)$.
5. Finally, it must ensure and notify that the completion has been finished and markings have been updated by the gates accordingly, so that the rest of the activities can evaluate the new enablement states.

The SAN formalism does not specify a completion order of the multiple input and output gates. However, it does specify that input gates must be completed before output gates. Therefore, caution must be taken when modeling a system, given that a different selection of the execution order of the multiple gates may cause different outcomes. However, this is not a common situation, and multiple gates can be merged into a single gate with a more complex function-set to avoid ambiguities. Furthermore, all the input gates can be executed in parallel and later the output gate set, if their executions do not alter the same places simultaneously and the gate functions do not depend on the markings that are altered by the rest of the input/output gates. This constraint is commonly met in SAN during model design and otherwise the models can be adapted so that this constraint is met to prevent ambiguities.

Formally, denoting $\text{OG}(ac_k)$ and $\text{IG}(ac_k)$ the sets of output and input gates connected to activity ac_k, og_n and ig_m the n^{th} output gate and m^{th} input gate of ac_k respectively, and $Pl(g)$ the set of places connected to gate g, the execution of any given input and output gates can be parallelized in HW for any gate functions and predicates, if and only if (Garro et al., 2019b):

$$\forall og_i,og_j \in \mathrm{OG}(ac_k) : i \neq j,\ Pl(og_i) \cap Pl(og_j) = \emptyset \tag{4.9}$$

$$\forall\, ig_i,ig_j \in \mathrm{IG}(ac_k) \ : i \neq j,\ Pl(ig_i) \cap Pl(ig_j) = \emptyset \tag{4.10}$$

Notice that input and output gates cannot be executed simultaneously. They must be sequenced, because frequently they are employed in recurrent processes as shown in Figure 4.13

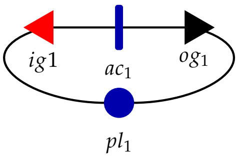


Figure 4.13: Recurrent SAN activity example.

If Equations (4.9) and (4.10) are met, the activity can complete in just two stages by requesting the parallel completion to the multiple input or output gate blocks with a `trig` signal. When input and output gate blocks receive `trig` signals, they read the markings of the associated places, operate their functions on the markings and write the values back, denoted with μ `i/o` in Figure 4.14. The activities await for all the gates to signal their completion with an `ack` signal and notify the completion to the scheduler at this point.

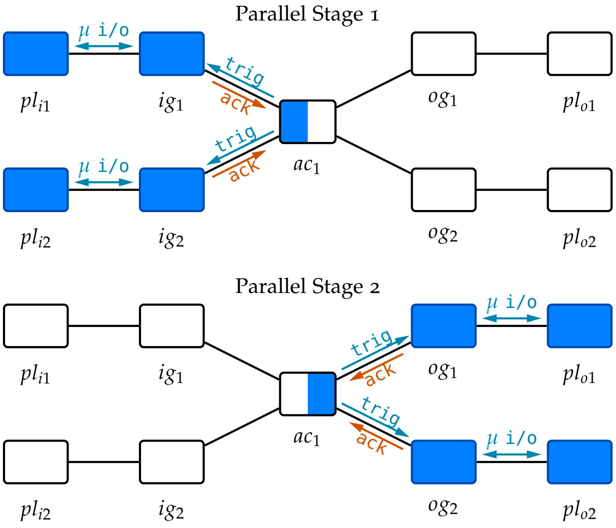


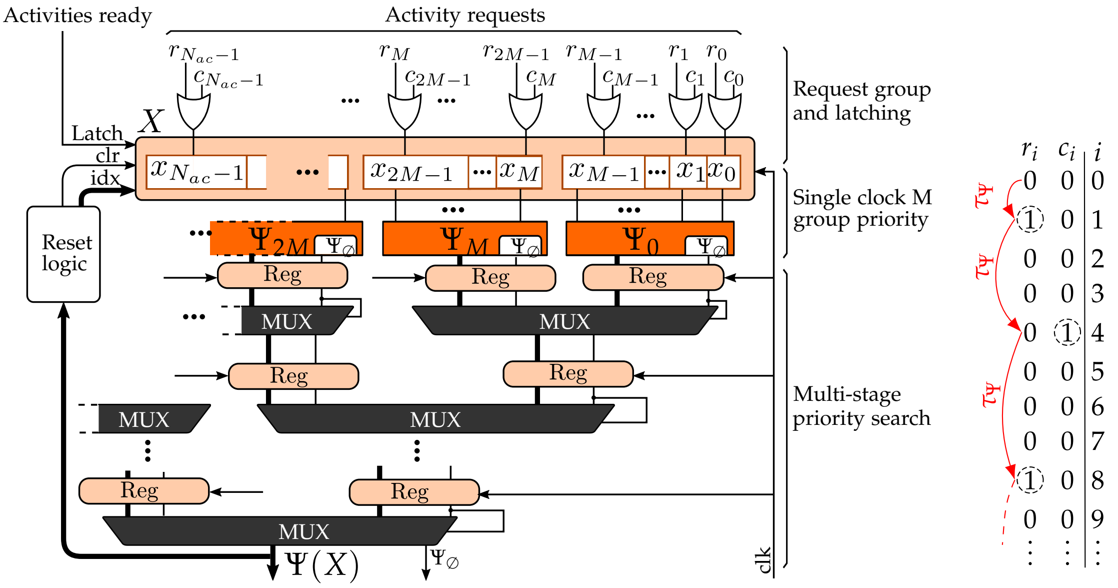
Figure 4.14: Stages of the completion of an activity.

4.5.1.3 *Accelerated activity request sorting*

The scheduler for a SAN simulation needs to update the completion order of activities at every activity completion. Every time the marking of the SAN model μ_{SAN} is altered, certain activities will be disabled and others enabled. A sequential collection of states of the activities as described in the flowchart in Figure 4.11 can be inefficient for larger models. The sequential collection would require every activity to be sequentially evaluated in Conditions ⓑ and ⓓ and Processes ⑥ and ⑦, requiring a collection time of $\tau_{\mathrm{coll}} = N_{ac}$ clock cycles at every activity completion.

On the other hand, a parallelization of the full activity state collection and PRNG would require large FPGA resources (see Section 4.4.4). However, in practice even larger SAN models have a low average number of activities n_{req} simultaneously requesting the generation of PRNG or their cancellation. This condition permits accelerating the collection of requests (Conditions ⓑ and ⓓ) while sequencing the PRNG (Process ⑦).

A HW implementation can be utilized for the parallel searching of requests, in order to accelerate the activity state collection. The architecture is based on the fast request search architecture shown in Figure 4.15a, that employs multiple priority encoder blocks. Priority encoders are combinational circuits, commonly used in logic circuits for highest priority interrupt selection, that output the identifier corresponding to the input with the highest priority among the inputs that are enabled. In this implementation, the highest *priority* is defined as the activity identifier with the lowest number (Short, 2009).



(a)

(b)

Figure 4.15: Accelerated activity request location architecture: (a) Pipeline architecture and (b) searching sequence example.

In this design, activity requests for new PRNG intervals r_i or their cancellation requests c_i, are first collapsed into single x_i activity requests that are recorded into an X register array of length N_{ac}. This array is grouped into M-sized X_M request sets that permit the operation of the priority encoders at higher clock frequencies. Each priority encoder block searches for the first request in a set X_M of M activity requests, and the output is given in a single clock. In this design a second output was provided to each of the priority encoder blocks denoted $\Psi_\varnothing$, that indicates no requests were found, that is, it outputs the result of the negated logical OR operation on all input requests. Each priority encoder Ψ_o operates on the o^{th} request-set $X_o = \{x_{oM} \dots x_{(o+1)M-1}\}$ and it is parameterized to produce outputs that reflect its offset position o:

$$\begin{aligned} U &= \{u_0 \dots u_{M-1}\},\ u_i \in \{0,1\} \\ \Psi_o(U) &:= \min\{m \in \{0 \dots M-1\} | u_m = 1\} + o \\ \Psi_\varnothing &:= \neg \bigvee_{i=0}^{M-1} u_i \end{aligned} \tag{4.11}$$

Once the priority encoders output the first level of the priority values, each of the next stage multiplexers selects the request with the highest priority (the one with the lowest activity index).

After each activity completion, this structure produces the first activity ac_i found to be requesting either a new PRNG number generation (r_i), or a cancellation due to its disablement (c_i). Then the activity is attended, either by assigning it a new interval by the PRNG, or by canceling the previous scheduled completion time. This sequence is repeated until all activity requests have been fulfilled (`req` signals cleared), as shown in Figure 4.15b. Using this method, activities with indexes in between those with requests are not processed, unlike the sequential search method and it reduces the required collection time.

The architecture is scaled to larger SAN models by the multi-stage multiplexer set shown in Figure 4.15a. Notice that the architecture permits pipelining and therefore it can be left in operation in parallel with the rest of the scheduler components. However, it does not benefit from the mechanism because the generated outputs will only be valid when the activity request inputs are valid, that is, when the inputs when activities are in ready state, and only after the pipeline latency has elapsed. The latency of the full pipeline ltn_Ψ grows logarithmically with the SAN model:

$$\mathrm{ltn}_\Psi = \left\lceil log_2 \left\lceil \frac{N_{ac}}{M} \right\rceil \right\rceil + 1 \tag{4.12}$$

At each SAN activity completion a generally low number of $n_{ac\mathrm{req}}$ activity requests are produced. Therefore, the average of the number of clock cycles to attend all activity requests in each completion $\tau_{ac\mathrm{att}}$ is

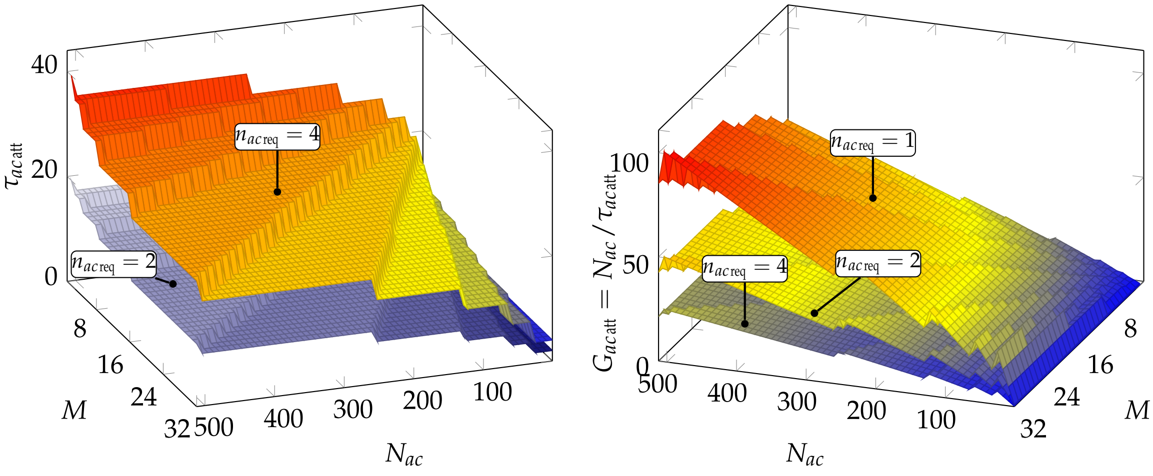
$$\mathrm{E}\left[\tau_{ac\mathrm{att}}\right] = n_{ac\mathrm{req}} \tau_\Psi = n_{ac\mathrm{req}} \left(\left\lceil log_2 \left\lceil \frac{N_{ac}}{M} \right\rceil \right\rceil + 1 \right) \tag{4.13}$$

where N_{ac} is the number of activities in the SAN model and M is the input size of the priority encoder blocks. As shown in Figure 4.16a, $\tau_{ac\mathrm{att}}$ grows logarithmically with the size of the model N_{ac}, independent of the complexity of the model, and it only grows linearly with the number of activities activated at each completion, which is often a single or two activities by design. While the latency can be further reduced by increasing the size of the priority encoders M, it can be observed that the latency also decreases logarithmically with respect to this size M. Therefore, the resource cost-benefit ratio decreases as M grows.

The latency reduction or speed gain in comparison to a sequential search of the activity requests $G_{ac\mathrm{att}} = \frac{N_{ac}}{\tau_{ac\mathrm{att}}}$ is plotted in Figure 4.16b. It can be observed that the benefit increases as the model size grows, with up to 96 times speed gain in the case of $n_{ac\mathrm{req}} = 1$ in comparison to a sequential search with the model sizes and HW complexities considered. However, it can be also observed that the speed gain increase rate (its derivative) also decreases as the model size increases.

4.5.1.4 *Parallelization of the next activity evaluation*

The evaluation of the next activity to be completed ac_{comp} and its completion time $\tau_{ac_{\mathrm{comp}}}$ requires evaluating the activity with the minimum completion time in the completion time list of the scheduler Ct, and the corresponding index number. This minimization problem can be solved by a pipelined parallelized architecture denoted $\mathcal{M}$. In theory, a combinational block designed to produce the output directly in a single clock cycle. However, latency requirements are not easily met for larger models due to the complexity of the minimization algorithm, and employing a sequential search in segments of size J and the



(a) (b)

Figure 4.16: (a) latency of the request location method for different average activity request numbers and (b) obtained gain in comparison to a sequential search, as a function of model size N_{ac} and priority encoder size M.

subsequent single clock minimum output would only lead to a $\mathrm{ltn}_{\mathcal{M}}$ minimization latency of

$$\mathrm{ltn}_{\mathcal{M}} = \left\lceil \frac{N_{ac}}{J} \right\rceil \tag{4.14}$$

The alternative is segmenting the input completion time list into segments of size J and producing the minimum value of each segment. Then the outputs are grouped again into segments to produce the next stage values, until the minimum value of the full set is evaluated. Figure 4.17 shows an example implementation for J = 2, where the minimization algorithm for the subsets is simplified as a comparator and multiplexer.

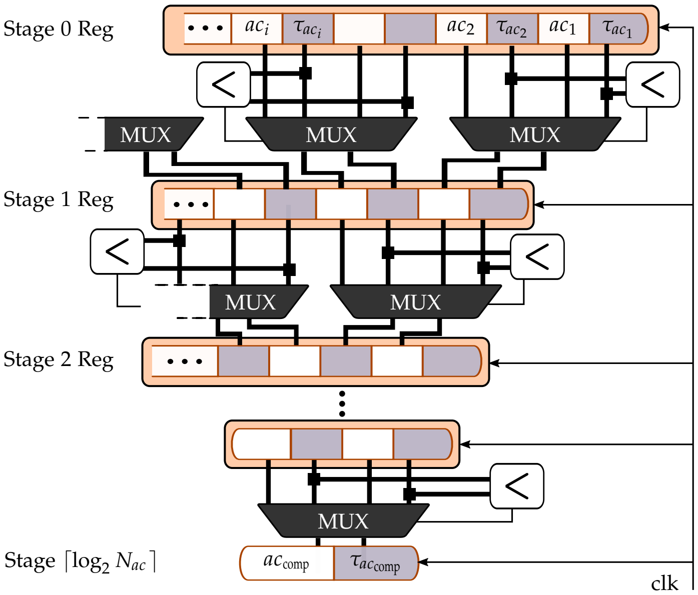


Figure 4.17: Pipelined parallel architecture that evaluates the next activity to be completed and its completion time, for J = 2.

In general, the latency $\mathrm{ltn}_{\mathcal{M}}$ of the pipeline in Figure 4.17 is

$$\mathrm{ltn}_{\mathcal{M}} = \lceil log_{\mathrm{J}} N_{ac} \rceil \tag{4.15}$$

The architecture can be left running in parallel without any synchronization with the scheduler, given that it is stateless, and therefore invalid inputs do not affect the results for the next input set. Notice, however, that in practice the inputs can depend on the current marking, and therefore they are unavailable until each activity completion is executed and propagates the markings completely. Therefore, the full latency of the pipeline needs to be elapsed until the next activity completion time is obtained.

It should be noted that this architecture is heavily parallelized and therefore it can consume large resources. However, the number of comparator plus multiplexer blocks required is:

$$N_{\mathrm{cmux}} \leq \sum_{i=0}^{\lceil \log_{\mathrm{J}} N_{ac} \rceil - 1} \mathrm{J}^i = \frac{1 - \mathrm{J}^{\lceil \log_{\mathrm{J}} N_{ac} \rceil}}{1 - \mathrm{J}} \leq \frac{1 - \mathrm{J}N_{ac}}{1 - \mathrm{J}} \tag{4.16}$$

The result from Equation (4.16) is an upper bound because the number of activities is not necessarily a power of J value or proportional to J, leading to optimizing out multiple blocks through one or multiple layers in the pipeline. This upper bound grows linearly with the SAN model size (measured by the activity count N_{ac}). The actual resulting resource usage of an implementation for a XCZU9EG-2 Xilinx FPGA is shown in Figure 4.18.

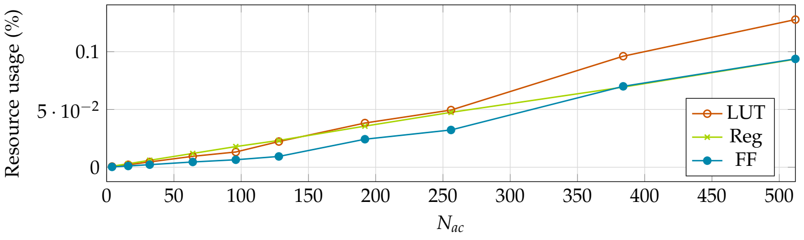


Figure 4.18: Resource usage of the minimization algorithm.

4.5.1.5 *Parallel evaluation of gate functions and input gate predicates*

After an activity completes, the new marking-set of the SAN model μ_{SAN} alters the enablement state of the activities. With the architecture described in Figure 4.10, it is evident that the evaluation of the predicate functions for the given marking can be executed in parallel. Therefore, all input gates are evaluated in parallel. Their state, however, cannot be updated unless the markings of the places are up-to-date. Any transition values of the markings must be avoided.

Due to the distributed nature of the HW simulation architecture, an activity block lacks of the notion of the transitioning state of the markings, and this condition can solely be evaluated by the scheduler. This state is controlled in Process ⑩ in Figure 4.11. The scheduler receives the completion state of the activity that was just triggered with an acknowledgment `ack` signal, indicating that the marking has propagated, and then it distributes this condition to the rest of activities.

If the predicate functions of the input gates of the full SAN model are simple logical and stateless operations, and they operate with a fixed latency (typically a single clock cycle), there is no need to distribute the completion signal to all other activities. In this case, the scheduler can just wait for the required clock cycles for the activities to establish the new enablement state.

However, the functions of the input and output gates can require different latencies depending on their complexity. Therefore, for a general solution, an acknowledgment mechanism is necessary between the activities and the gates, as described in Figure 4.19.

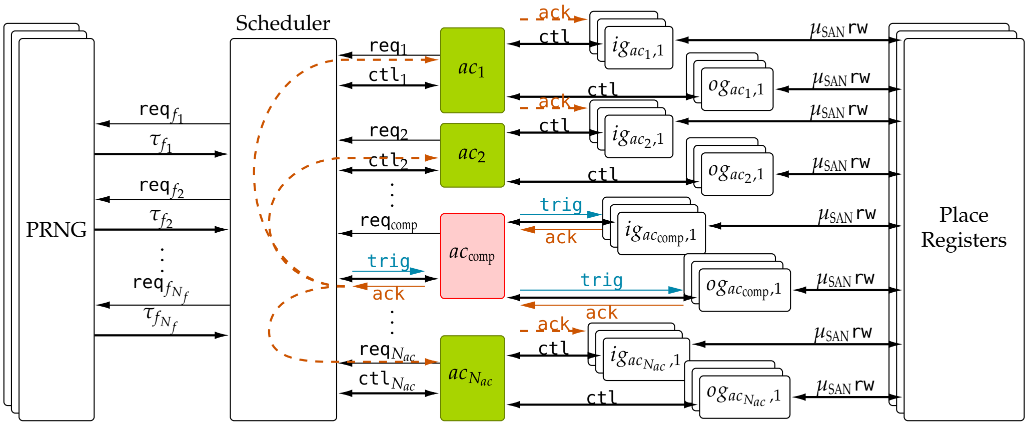


Figure 4.19: Distribution of marking propagation ready signal to all activities.

Additionally, as described in Section 4.5.1.2, the input and output gate functions are triggered in sequence (output gates after input gates), but all input gates can be triggered in parallel, and also all output gates, presuming that conditions Equations (4.9) and (4.10) are met. These conditions are generally assumed for properly designed models and avoid ambiguities, and they are met in the existing literature and considered to be met in existing simulation SW such as Möbius (Sanders and Meyer, 2001; *The Möbius Tool* 2017). Therefore, these conditions have also been assumed for the implementations throughout this thesis.

The marking propagation acknowledgment mechanism of Figure 4.19 is implemented in the activity blocks as described in the flowchart in Figure 4.20.

In this flowchart, the activity blocks begin by initializing their outputs (Process ①) and waits until all the input gates are ready to evaluate their enablement state (Condition ⓐ). Then a new random number is requested (Process ②) and wait until the scheduler requests for a completion of the activity by triggering it (Condition ⓒ). At this point, any other activity or itself can be requested to complete and alter the marking of the places. If complex predicate functions are stateful or they require multiple or varying clock cycle latencies, it is necessary to check for the marking propagation to be finished (Condition ⓓ), before evaluating if the activity has been disabled (Condition ⓔ). Otherwise the marking propagation step can be avoided and the scheduler can take care of the transitions of the markings on its own without any propagation of the acknowledgements described in Figure 4.19. If the activity is disabled by the completion of another activity, its cancellation is requested Process ⑥. On the other hand, if triggered (the scheduler requests its completion in Condition ⓒ), input and output gates are triggered in sequence (Processes ⑤ and ⑦).

The number of clock cycles $\tau_{\text{act}}(ac_k, t_{\text{SIM}})$ required to process the completion of a given activity ac_k at a given simulation time t_{SIM} can be bounded, if the maximum of the latency set of its connected output gate set at this simulation time instant $\tau_{\text{OG}}(ac_k, t_{\text{SIM}})$ and the maximum of the input gate set latencies $\tau_{\text{IG}}(ac_k, t_{\text{SIM}})$ can be bounded:

$$\tau_{\text{act}}(ac_k, t_{\text{SIM}}) = \max\{\tau_{\text{OG}}(ac_k, t_{\text{SIM}})\} + \max\{\tau_{\text{IG}}(ac_k, t_{\text{SIM}})\} + 3 \tag{4.17}$$

where 3 corresponds to the number of clock cycles required for the signaling and acknowledgment process. This latency $\tau_{\text{act}}(ac_k, t_{\text{SIM}})$ is deterministic if the maximum latency of the

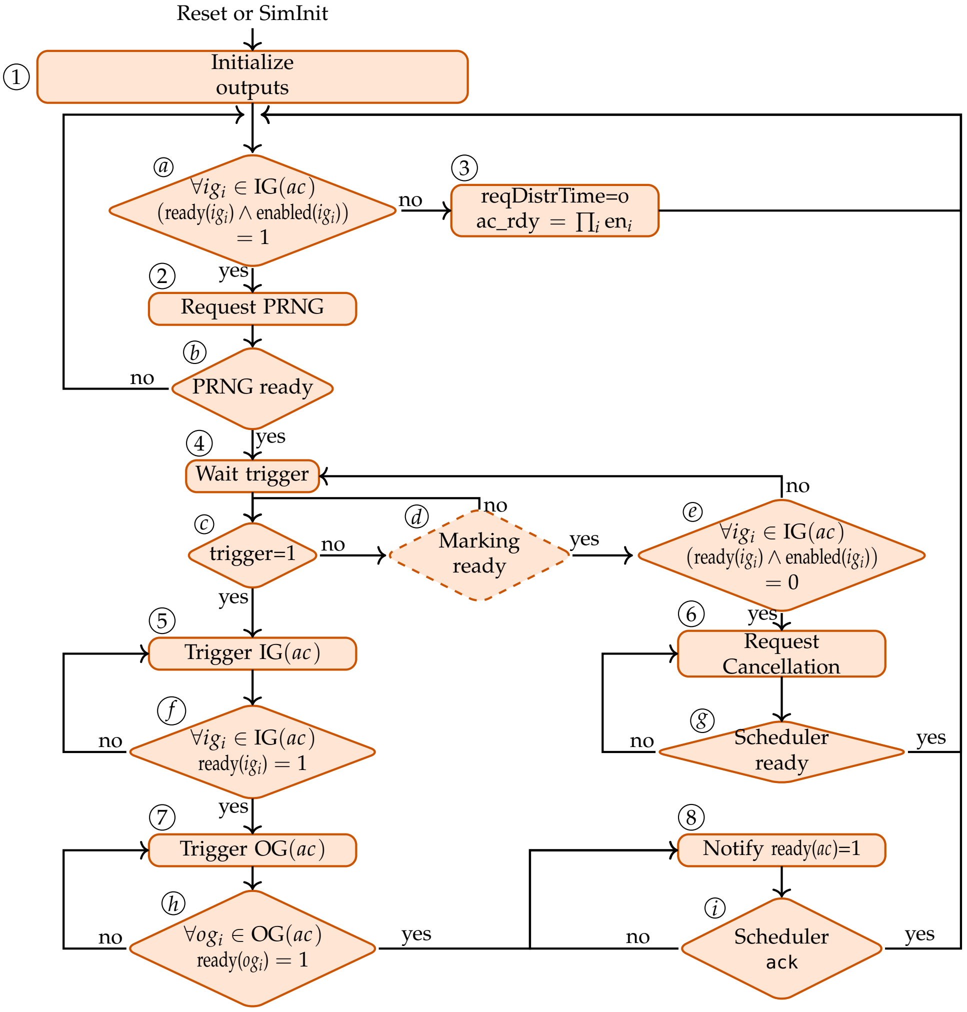


Figure 4.20: Flowchart of the state machine implemented by activity blocks.

gate functions remains constant throughout the simulation. Details on the execution cycles and actual signal implementations are provided in Figure A.7.

4.5.1.6 *Parallel simulation of the physical models*

The completion sequence is governed by the central scheduler. However, when certain activities involve physical model simulations, the scheduler needs the physical model execution to schedule the triggering of the resulting event. Therefore, in this case their completion time is generated by the activity block. However, other activities may complete earlier than the physics-based model activity. Thus, this activity simulation is limited by the minimum completion time of all other activities in the model $\tau_{ac_{min}}$ (see Section 2.3.2), which is reported by the scheduler as a mission time limit $t_{M,\text{Phys}}$ to the physical model activity. As described in Figure 4.21 flowchart, in order to adapt this strategy to the distributed architecture, the activity block must still notify the generation of a new completion time to the scheduler (Process ②). If the activity block is a physical model, the scheduler will request a new *self-generated* (and random) interval to be generated by the physical model (Condition ⓒ). Then, the parallel simulation of the physical model begins (Process ④) and the resulting completion time $\tau_{ac_{phys}}$ is reported to the scheduler (Equation (2.6)).

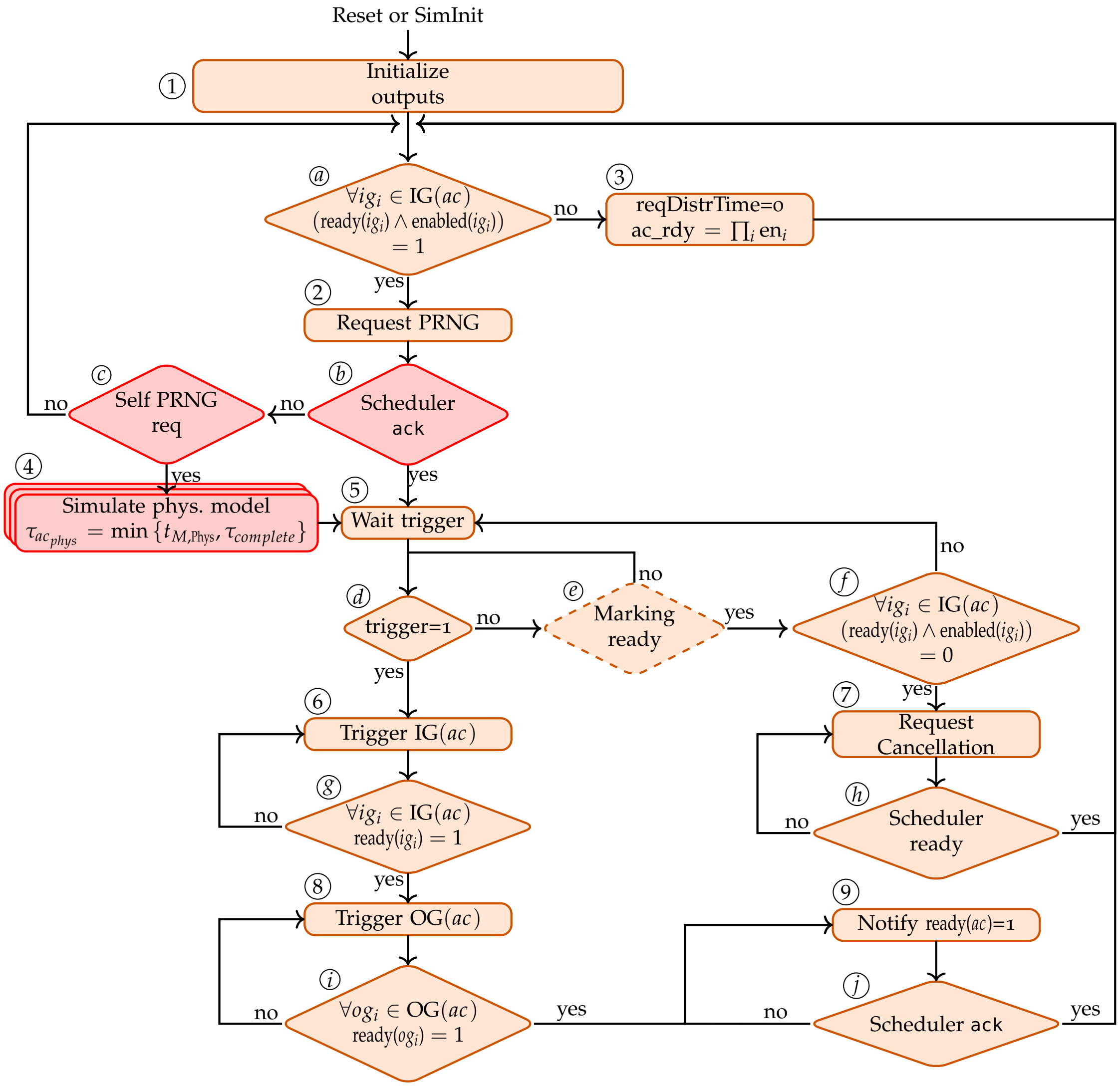


Figure 4.21: Flowchart of the state machine implemented by activity blocks, supporting physical models.

The scheduler implements this sequence as shown in Figure 4.22. In this modified flowchart, when an activity implements a physical model, it is reported to the scheduler as a specific distribution type (Condition ⓒ). Accordingly, the completion time of the rest of activities is reported to the activity and once ready a request is produced to process

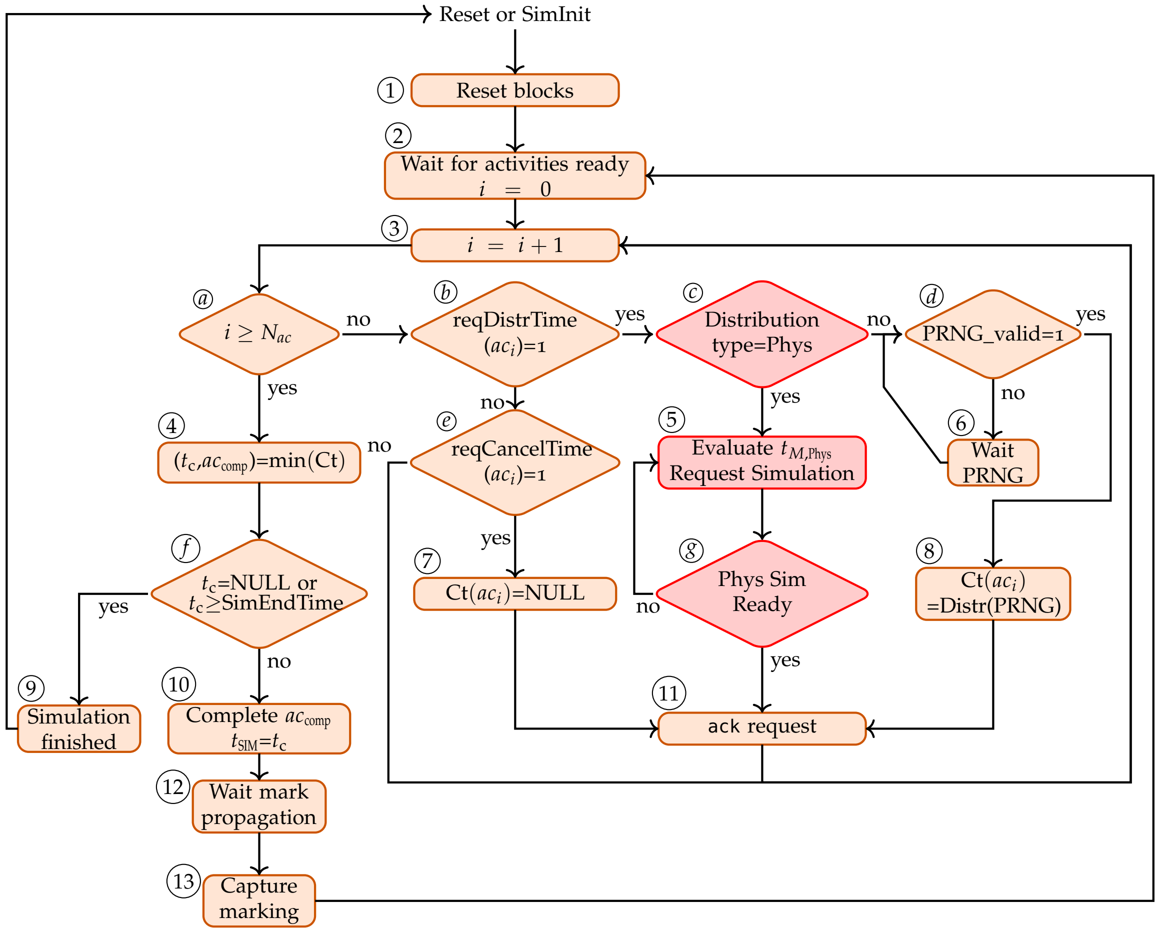


Figure 4.22: Flowchart of the state machine implemented by the scheduler, supporting activities with physical models.

the simulation, limited by this mission time $t_{M,\text{Phys}}$ (Process ⑤). The simulation is then performed in parallel by the activity block. Once ready (Condition ⓖ), the simulation is acknowleged (Process ⑪). Notice that the physical simulations depend on the completion time of the rest of activities. Therefore, as discussed in Section 2.3.2 this activity ac_{Phys} must be the last activity block processed by the scheduler. This can be achieved by defining this activity as the last activity connected to the scheduler or the activity with the largest identifier ($ac_{N_{ac}}$) as shown in Figure 4.23

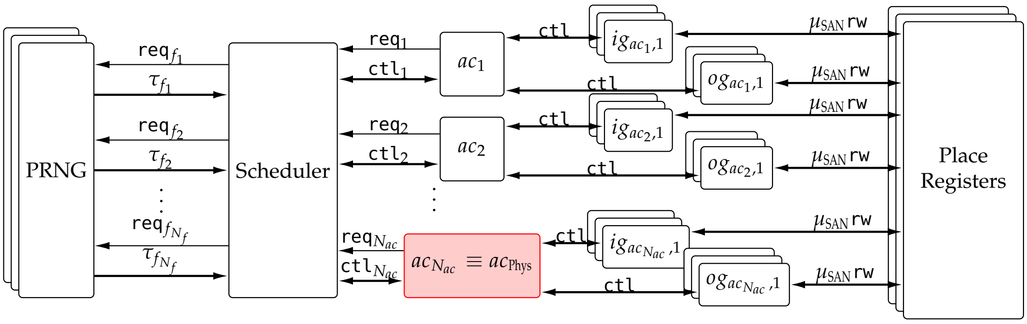


Figure 4.23: Topology for a distributed component FPGA SAN simulator including an activity driven by a physical model.

4.5.1.7 *Self Generator of Random Numbers and Marking Dependent Distributions in Activities*

Certain activities may require the generation of random numbers, related to the stochastic nature of the processes and unrelated to their completion time intervals. Examples of such processes can be sensor noise, random grid load values or the random outcome of a component behavior. These processes may require additional PRNG to be integrated within the activity blocks. The same structures described in Section 4.4.3 can be employed for this purpose. An example of the implementation for these processes is described in the cable model employed in through this thesis in Chapter 5.

The completion times of some activities may also alter its PDF parameters depending on the current marking. Example scenarios of this requirement are the switching times of a circuit breaker depending on the load factor (Mehl and Meckler, 2005), or the varying degradation model of a cable for a given load factor. This situation implies that an activity needs to have the possibility to monitor the marking of the places, which can be achieved simply by the architecture shown in Figure 4.24.

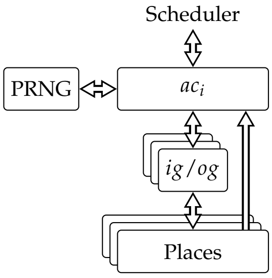


Figure 4.24: Self-generated PRNG and marking dependent distribution implementations.

4.5.1.8 *Parallel Execution of the SAN Simulation*

Figure 4.25 shows the full SAN simulator architecture. Notice that each gate can be connected to multiple activities. In this figure crossed connections between activities and gates have not been drawn for clarity of the drawing. The possible PRNG blocks required by the activities have also been integrated within the activity itself. A sequential implementation of the state machine described by Figure 4.26 is implemented in the FPGA block ❶ that controls the simulation of the SAN model and provides a global timing and sequencing reference. This state machine collects the activity states in Process ③ and records all their requests in array ❷ when they are ready. Process ④ employs the priority encoder described in Section 4.5.1.3 to obtain the next request to be processed. The priority encoder architecture (see Figure 4.15a) is implemented by block ❹. When no requests are available, a NULL value is returned by the architecture. Notice that only those activities with requests for the generation of new random numbers or their cancellation are reported. Therefore, one of the conditions of the flowchart (the request for cancellation indicated in dashed lines) is no longer necessary in the state machine.

All the completion times are generated by the scheduler in Process ⑨, that employs PRNG block ❾. Once the array of completion times Ct is ready in block ❸, block ❺ obtains the next activity to be completed, which is implemented with the architecture described in Figure 4.17. The block returns the next activity to be completed ac_{comp} and its completion time $\tau_{ac_{\mathrm{comp}}}$.

At every activity completion, the current simulation time t_{SIM} is updated and reported to the result collector ⓫. A `Latch` signal is issued, requesting the collector to capture the current state. This block checks and records every state change in the marking of the connected places. Once the simulation reaches its configured mission time t_M, the scheduler

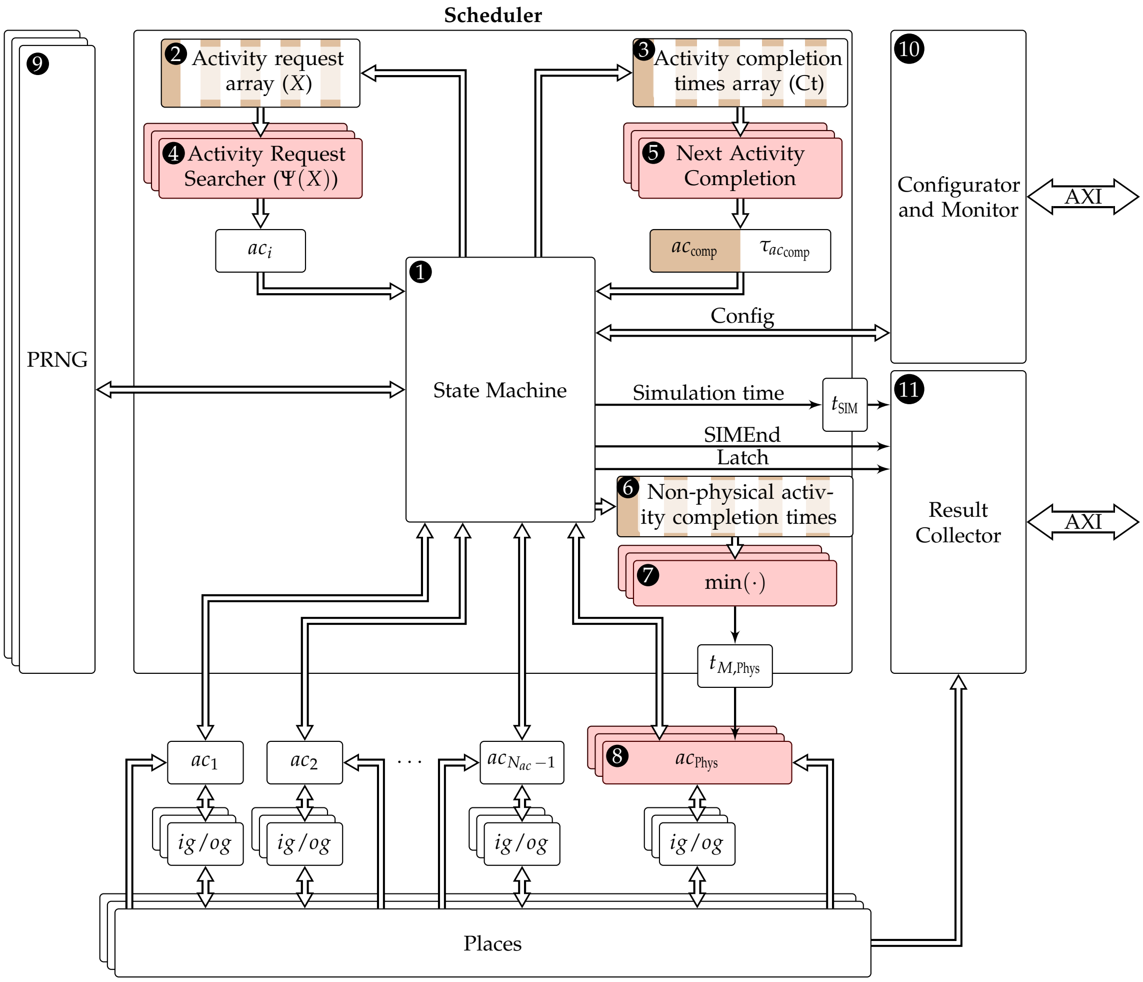


Figure 4.25: Block diagram for the parallel FPGA SAN simulator architecture.

notifies the event with signal `SIMEnd` (Process ⑩) and the result collector transfers the results to main memory over the AXI bus. This process is described in Section 4.7.

Block ❻ contains a duplicate of the Ct array excluding the completion time of the physical activity. This block is required to obtain the partial mission time issued to the physical activity $t_{M,\mathrm{Phys}}$, as described in Section 4.5.1.6. The process to obtain the mission time employs block ❼ the same architecture from Figure 4.17 as with block ❺.

The *configurator and monitor* block ❿ is an AXI slave block that permits configuring and monitoring the simulation process, and configuring parameters of the activities (including the physical models), with direct connections. The possible configurations include the simulation mission time t_M and the number of MC simulations to be performed N_{MC}. It permits monitoring the internal behavior of the simulation, including the number of simulations performed *SimCount*, the number of activity completions executed and the current simulation time t_{SIM}. Once configured, the scheduler executes all the MC simulations in sequence until the number of simulations requested is met (Condition ⓖ).

4.6 Latency of the SAN simulator

While the simulation of rare events requires a large number of simulations, the simulation of frequent events (usually operational events) in a SAN model can require the completion of thousands or millions of activities in each simulation. In order to reduce the simulation time of each MC trial, it is desirable to reduce the total activity execution time. The

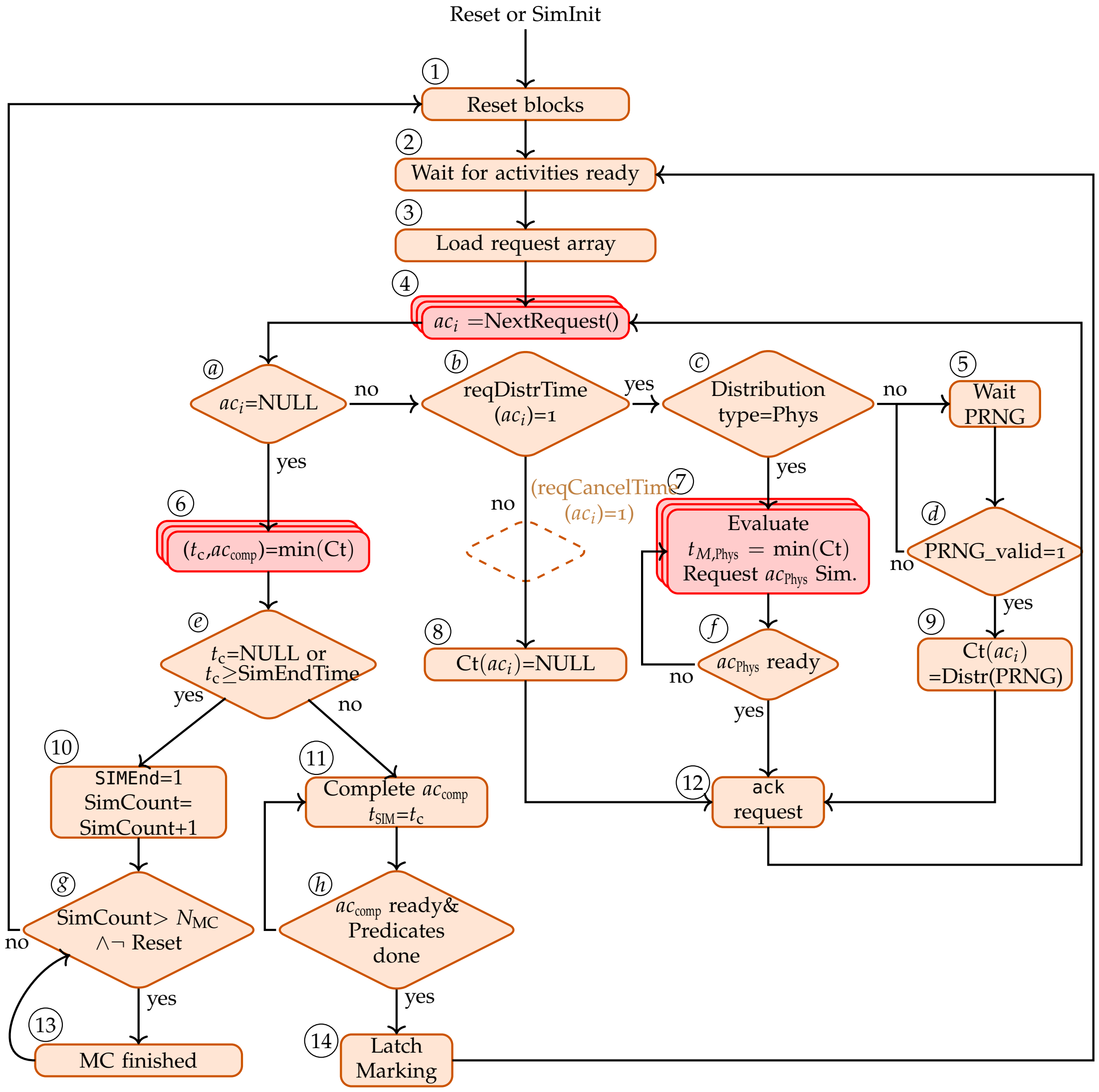


Figure 4.26: Flowchart diagram for a parallel SAN simulation.

process involves the completion of the activities, the propagation of the markings and the updating of the scheduler sequences. Identifying as ac_k the k^{th} activity triggering in any SAN completion sequence, if ac_j gets enabled after an activity ac_i completes at simulation time $t_{\text{SIM}} = t_i$ as shown in Figure 4.27, the number of clock cycles $\tau^{(ac_i)}_{\text{trig},ac_j}(t_i)$ required by the scheduler to trigger (and complete) ac_j after ac_i (if it is not canceled) is dependent on the model state at time t_i:

$$\tau^{(ac_i)}_{\text{trig},ac_j}(t_i) = \sum_{k=i+1}^{j} \tau^{(ac_{k-1})}_{\text{trig},ac_k}(t_k) = \sum_{k=i+1}^{j} \left(\tau_{ac\text{att}}(t_k) + \text{ltn}_{\mathcal{M}} + \tau_{\text{act}}(ac_k, t_k) + \tau_{\text{Sch}}(t_k)\right) \tag{4.18}$$

The time to attend activity requests at each simulation time $\tau_{ac\text{att}}(t_k)$ is variable in time, but generally it is constant after the completion of a given activity. Its expected value can be obtained from Equation (4.13). The minimization algorithm latency $\text{ltn}_{\mathcal{M}}$ is constant and given by Equation (4.16) and the completion time for each given activity is generally constant, although it may vary with time for complex gate functions and activity distributions. $\tau_{\text{Sch}}(t_{\text{SIM}})$ denotes the delay caused by the scheduler switching and acknowledgement mechanisms with the rest of blocks, excluding activities, because this 3 clock cycles delay is included within the activity completion time. This last term can be minimized by remov-

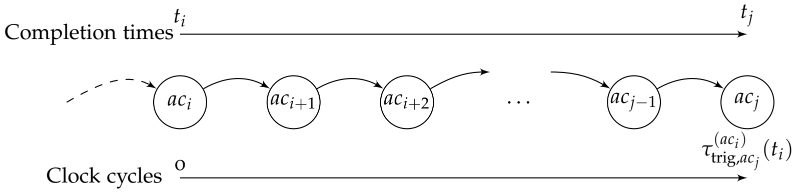


Figure 4.27: Completion sequence and execution clock cycles from any two given activities ac_i and ac_j.

ing acknowledgement mechanisms, if the function latencies in the gates, the predicates, completion times and PRNG are designed to be deterministic. However, the removal of the acknowledgement signaling invalidates the generalization of the method.

Therefore, the simulation time for a given sequence can be estimated as follows:

$$\begin{aligned}\mathrm{E}\left[\tau^{(ac_i)}_{\mathrm{trig},ac_j}(t_i)\right] = (j-i)\Bigg[& n_{ac\,\mathrm{req}}\left(\left\lceil log_2\left\lceil\frac{N_{ac}}{M}\right\rceil\right\rceil+1\right)+\lceil log_J N_{ac}\rceil \\ &+\mathrm{E}\left[\max\{\tau_{\mathrm{OG}}(ac,t_{\mathrm{SIM}})\}+\max\{\tau_{\mathrm{IG}}(ac,t_{\mathrm{SIM}})\}\right]+3+\mathrm{E}\left[\tau_{\mathrm{Sch}}(t_k)\right]\Bigg]\end{aligned} \tag{4.19}$$

It can be observed that, for models with constant input and output gate latencies and scheduler dynamics, the simulation time grows logarithmically with the model size. Therefore, its efficiency in comparison to sequential methods increases with the model size.

4.7 THE REWARD COLLECTION PROCESS

The rewards of the SAN model can apply functions on the markings of the SAN model. The same result can be obtained by applying an output gate function to a place and reading out its marking value. Accordingly, the architecture shown in Figure 4.25 uses this method. The collection of the markings is performed by the *Result Collector* block of the FPGA architecture.

Collecting the state of any of the places in a SAN model can require large memory resources for larger mission times and very frequently varying markings. Storing all the results in memory can require large memory transfers to RAM memory, impacting on the efficiency of the CPUs and causing delays in the simulation process. However, in high reliability equipment such as electric power distribution grid, the frequency of occurrence of the events of interest is low. In this context, the developed collection method takes advantage of the rarity of the events that cause the marking changes of the places indicating system or component failures.

This characteristic enables the system to collect just a small number of records per simulation, which can be stored in FPGA registers for higher efficiency, and transferring the results to RAM memory only when the simulation finishes. In this context, the collector architecture, shown in Figure 4.28, consists of a configurable number of N_{reg} registers, where each register records the markings of multiple places of the SAN model, and additionally it includes the time at which the marking change occurred. A round-robin logic controls that data is latched into a new register every time the scheduler issues a `Latch` signal. However, the latch logic only permits recording the new value when the new markings differ from the past recorded markings. The collector is configured to be an AXI bus master, and the records are transferred to a preconfigured RAM memory address after the scheduler indicates the end of each simulation with the signal `SIMEnd`.

Figure 4.29 shows the format of the results as organized in RAM memory.

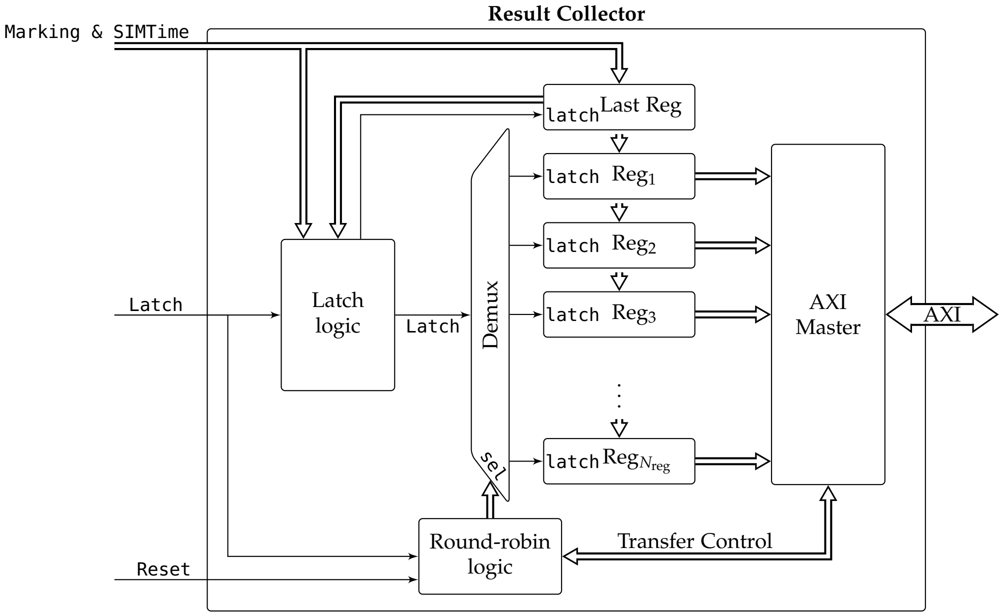


Figure 4.28: Architecture of the result collector.

Record: first row of the table below. Simulation: from the first row down to the t_{NLatch} row.

SIM Start	$Reg_1(0)$	$Reg_2(0)$	…	$Reg_{N_{reg}}(0)$
t_1	$Reg_1(t_1)$	$Reg_2(t_1)$	…	$Reg_{N_{reg}}(t_1)$
t_2	$Reg_1(t_2)$	$Reg_2(t_2)$	…	$Reg_{N_{reg}}(t_2)$
		⋮		
t_{NLatch}	$Reg_1(t_{NLatch})$	$Reg_2(t_{NLatch})$	…	$Reg_{N_{reg}}(t_{NLatch})$
SIM Start	$Reg_1(0)$	$Reg_2(0)$	…	$Reg_{N_{reg}}(0)$
		⋮		

Figure 4.29: Memory format of the results

4.7.1 *Conditions to End a MC Trial*

During a simulation, when no activities are enabled, the completion time array is empty and a NULL value is returned for the next completion time, effectively ending a simulation. However, if not taken care of, a general SAN model can have activities that continue being enabled and complete even after a system failure occurs. In order to accelerate the MC simulation, it is sufficient to either:

a) Disable all activities in the SAN model whenever a system failure occurs. Or, if the process is complex due to the large SAN models,

b) add a marking condition to end the simulation in the state machine of the scheduler (Condition ⓔ in Figure 4.26).

4.8 CONFIGURATION, MONITORING AND POST-PROCESSING OF THE SIMULATION RESULTS

The presented architecture records all the reward changes and their timestamps in a given section of the RAM memory. The reward collection method is optimized to reduce the required memory transfer size (see Section 4.7). The number of simulations to be performed can be configured by the CPU through the configurator block, and whenever necessary to reduce the variance of the results, the number of simulations can be increased to produce additional results.

For efficiency purposes, the post-processing of the results to generate a PDF or CDF can be performed in the FPGA by implementing an accumulator block in the result collector. However, this feature has not been implemented in this thesis, because the captured rewards may not reflect actual failure events, but other metrics of interest for the application. Therefore, the approach described in Section 4.7 has been used. The generated timestamps, displayed in Figure 4.29 need to be arranged into *bin*s and accumulated to produce a histogram following Algorithm 1.

Algorithm 1 Histogram binning process

```
SimCount $\leftarrow 0$
while SimCount $< N_{MC}$ do
    Record $\leftarrow$ nextRecord()
    lastBin $\leftarrow 0$
    while Record.Timestamp $\neq$ SIM_START do
        recordBin=$\lfloor$Record.Timestamp/binSize$\rfloor$
        for all bin$\in \{$lastBin$, \ldots ,$recordBin$\}$ do
            Histogram[bin]=Histogram[bin]+Record.Reg
        lastBin $\leftarrow$ recordBin+1
        Record $\leftarrow$ nextRecord()
    SimCount $\leftarrow$ SimCount+1
end
```

The algorithm parses each of the MC trials until the next trial is found by finding a SIM_START timestamp. All registers are loaded from each of the records, and binned according to their timestamp Record.Timestamp. The proccess continues until the number of simulations read through the *configurator and monitor* block is counted in SimCount.

Frequently, current architectures feature multi-core CPUs. The presented FPGA architecture, with very low AXI transfers has a low impact on the CPU memory access performance. In this context, the lower speed of post-processing versus FPGA simulation times benefit from post-processing the results in parallel with the concurrent simulation in FPGA. Additionally, depending on the SAN model size, multiple simulators can be synthesized on the same FPGA, as shown in Figure 4.30. The figure shows a triple simulator architecture with a 2-core CPU. The simulation records are split between the two processors and accumulated separately into separate histograms by each CPU.

Each of the simulators in hardware features a separate AXI Direct Memory Access (DMA) controller that records the registers into memory in parallel as shown in Figure 4.30. Notice that each MC trial simulation time t_{sim} varies with time with the complexity of the simulation. DMA transfer times t_{DMA} can also vary with time due to the varying number of records in each MC trial, and also because of the increased latency that can cause parallel transfers by the multiple simulators.

The CPUs monitor the number of simulations performed employing the *Configurator and Monitor* block through the AXI bus, and collect the results up to the last completed MC trial.

4.9 Obtaining confidence margins for the results.

Obtaining the confidence intervals according to Equation (2.11) requires calculating the point-wise variance of the collected reward data.

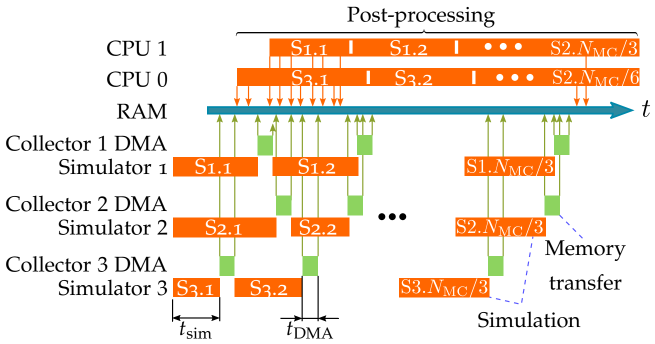


Figure 4.30: Concurrent post-processing of simulations.

The variance in the presented implementation is calculated by the CPU through post-processing the records. For general valued rewards obtained from the marking of each place pl_i at simulation time instant t of the j^{th} MC trial $\mu_i^{(j)}(t) = \mathrm{m}(pl_i, j, t)$, the expected value and variance of a marking after each trial j can be estimated recurrently without reprocessing all previous reward values, by (See Appendix A.4):

$$\operatorname{Var}\left(M_j\right) = \frac{\operatorname{Var}\left(M_{j-1}(t)\right)(j-1)}{j} + \frac{(\mu_i^{(j)}(t) - \bar{\mu}_i^{(j-1)}(t))^2(j-1)}{j^2} \tag{4.20}$$

$$\bar{\mu}_i^{(j)}(t) = \frac{(j-1)\bar{\mu}_i^{(j-1)}(t) + \mu_i^{(j)}(t)}{j} \tag{4.21}$$

where

$$M_{j-1}(t) = \{\mu_i^{(1)}(t), \mu_i^{(2)}(t), \ldots \mu_i^{(j-1)}(t)\} \tag{4.22}$$

$$M_j(t) = \{\mu_i^{(1)}(t), \mu_i^{(2)}(t), \ldots \mu_i^{(j)}(t)\} \tag{4.23}$$

$$\bar{\mu}_i^{(j-1)}(t) = \mathrm{E}\left[M_{j-1}(t)\right] \tag{4.24}$$

In reliability analysis, when the reward function is defined as the marking of a place pl_i that represents a system failure and its marking is defined as $\mathrm{m}(pl_i) \in \{0, 1\}$, the calculation of the variance is simplified as:

$$\begin{aligned}
\operatorname{Var}\left(M_j\right) &= \sum_{k=1}^{j}\left(\mu_i^{(j)}(t) - \bar{\mu}_i^{(j-1)}(t)\right)^2 \\
&= \sum_{k=1}^{j}\left(\left(\mu_i^{(j)}(t)\right)^2 - 2\mu_i^{(j)}(t)\bar{\mu}_i^{(j-1)}(t) + \left(\bar{\mu}_i^{(j-1)}(t)\right)^2\right) \\
&\overset{\mu_i^{(j)}(t)\in\{0,1\}}{=} \sum_{k=1}^{j}\left(\mu_i^{(j)}(t) - 2\mu_i^{(j)}(t)\bar{\mu}_i^{(j-1)}(t) + \left(\bar{\mu}_i^{(j-1)}(t)\right)^2\right) \\
&= \mu_i^{(j)}(t) - \left(\bar{\mu}_i^{(j-1)}(t)\right)^2
\end{aligned} \tag{4.25}$$

Equation (4.25) permits evaluating the variance sequentially and stopping the simulation when the required confidence interval is reached, according to Equation (2.11). The CPU can request additional simulations to be performed until the requirements are met.

4.10 ANALYSIS OF THE PERFORMANCE OF THE ARCHITECTURE: TRIPLE MODULAR REDUNDANCY (TMR) USE CASE

In order to test the implementation, performance and resource usage of the architecture, the following use-case has been analyzed.

The system to be modeled is shown in Figure 4.31. It represents an oil tank that randomly overheats, and a system controls a valve to control its temperature when such an event occurs.

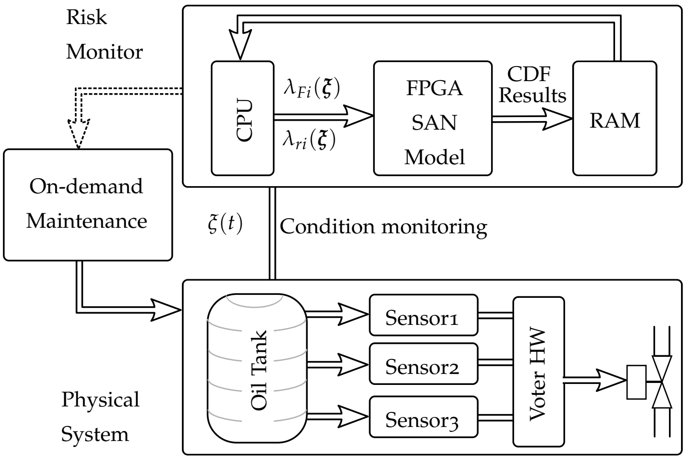


Figure 4.31: Hierarchical Stochastic Activity Network (HSAN) model of the overheating control system.

Due to the high criticality of the system, a TMR system has been put in place to control the valve. Three temperature sensors control the tank temperature. When the sensors indicate an overheating condition, the system acts on the valve. However, the sensors can fail and they are degrading in time and their reliability is monitored by a risk monitor. The CPU in the reliability monitor evaluates the degradation level set of the sensors ξ and outputs the failure PDF for each sensor. The degradation detection algorithm itself has not been modeled in this use case, but it is assumed that it produces exponentially distributed failure probabilities $\lambda_{Fi}(\xi)$ and repair rates $\lambda_{ri}(\xi)$ for each sensor, with varying failure rates for each.

The failures of the sensor are controlled by a voter mechanism that acts on the valve when at least two of the three sensors indicate an overheating condition. When a single sensor is in failed state, the voter is immune to the output of the failed sensor. However, when two or more sensors fail, the system is considered to be unavailable, as the voter is unable to tell if an overheating actually occurred.

The reliability monitor evaluates the transient values of the full system reliability $\mathcal{R}(t)$, given the monitored degradation level, which permits requesting or planning maintenance procedures.

These are the hypothesized system features and requirements:

- Includes repairable and redundant equipment.
- Includes both deterministic and stochastic time intervals.
- It requires long term prognostics.
- Failure causing events have of low probability of occurrence.
- Includes very frequently occurring events.

The hierarchical SAN model is shown in Figure 4.32. It includes macroactivity instances for each sensor failure models (*Sensor1*, *Sensor2* and *Sensor3*), the logic for the voter *TMRLogic*, and the model for the overheating and the valve control *OverheatingModel*. The instant activity $ac_{\text{OverheatingUndetected}}$ is activated when the system is unavailable due to 2 or more sensors being in failed state, and an overheating has occurred at the same time.

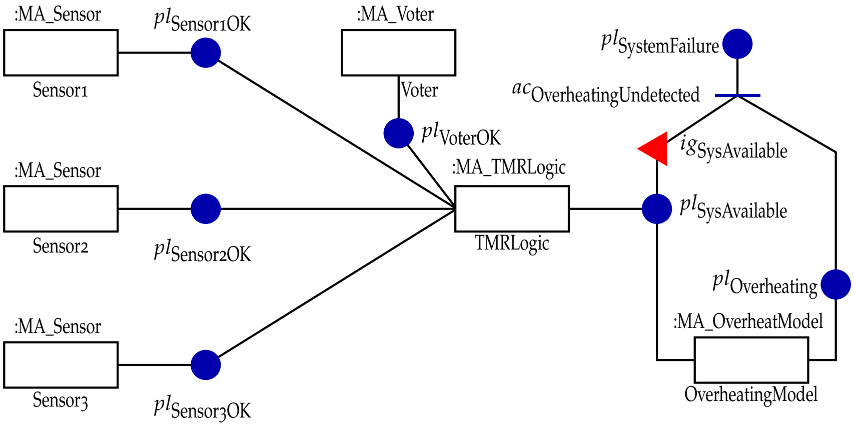


Figure 4.32: HSAN model of the overheating control system.

Figure 4.33 shows the model for the TMR logic macroactivity. The model switches between working and failed states ($pl_{\text{SysAvailable}}$ and $pl_{\text{SysUnavailable}}$) depending on the condition of the sensors. The sensor states are controlled by two input gates ($ig_{\text{SysAvailable}}$ and $ig_{\text{SysUnavail}}$). These input gates do not alter the markings of the connected places. Therefore, their input functions are null and they just implement the predicates that control the sensor states (pl_{Sens1OK}, pl_{Sens2OK} and pl_{Sens3OK}). The input gate functions and predicates of the model are included in Table 4.1. The table includes the C++ nemonic function (commonly used in SAN simulators such as Möbius Tool (*The Möbius Tool* 2017), and the equivalent VHDL function that can be employed for FPGA implementations. In the VHDL notation, MarkI refers to the input place marking for cases when a single input is available, and MarkS1,MarkS2,MarkS3 and MarkVT refer to the marking inputs corresponding to the block inputs for the sensors and voter condition, that are connected to pl_{Sens1OK}, pl_{Sens2OK}, pl_{Sens3OK} and pl_{VoterOK}.

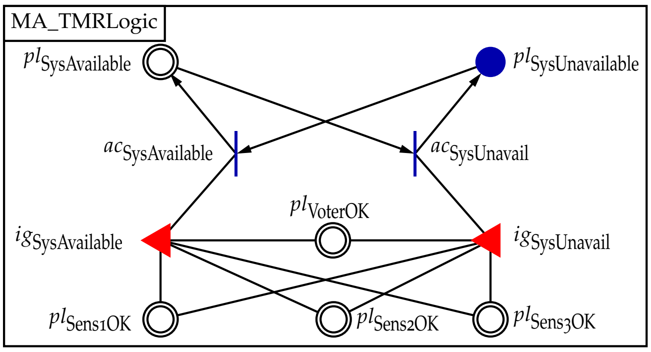


Figure 4.33: HSAN macroactivity model of the voter TMR logic.

Table 4.1: Input gate predicates. (All input gate functions are null.)

Input Gate	Predicate (C++)	Predicate (VHDL)
$ig_{\text{SysFailed}}$	m($pl_{\text{SysAvailable}}$)==0	**signed**(MarkI)= 0
ig_{SysOK}	m($pl_{\text{SysAvailable}}$)>0	**signed**(MarkI)>0
$ig_{\text{IGSysAvailable}}$	TMR^c(pl_{Sens1OK},pl_{Sens2OK},pl_{PSens3OK}) &&m(pl_{VoterOK})	TMR^v(**signed**(MarkS1),**signed**(MarkS2), **signed**(MarkS3)=1 **and** **signed**(MarkVT)=1)
$ig_{\text{SysUnavail}}$	!TMR^c(pl_{Sens1OK},pl_{Sens2OK},pl_{Sens3OK}) $\Vert$ m(pl_{VoterOK})!=1	TMR^v(**signed**(MarkS1),**signed**(MarkS2), **signed**(MarkS3))=0 **or** **signed**(MarkVT))=0)

TMR^c(x,y,z)=(m(x)&&m(y)) $\|$ (m(y)&&m(z)) $\|$ (m(z)&&m(x))

TMR^v(x,y,z)=((x=1) and (y=1)) or ((y=1) and (z=1)) or ((z=1) and (x=1))

The sensor model considers that each sensor can be in three different states:

(i) Faulted due to the degradation ($pl_{\text{SensorFaulted}}$);

(ii) being repaired by a maintenance procedure after a sensor ($pl_{\text{SensorStartup}}$);

(iii) fully operational (pl_{SensorOK}).

The fault and repair rates estimated by the CPU are applied to $ac_{SensorFault}$ and $ac_{SensorRepair}$, and the startup time in $ac_{SensorStartup}$ are considered to be deterministic and constant intervals.

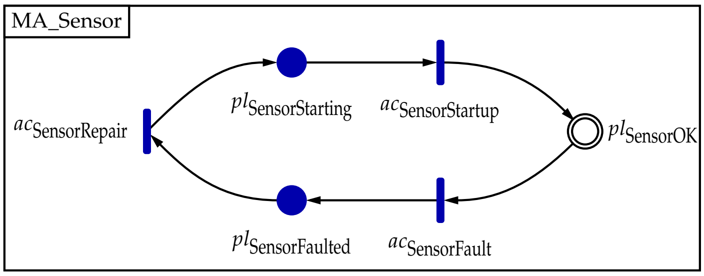


Figure 4.34: HSAN macroactivity model of the overheating sensor.

The voter model in Figure 4.35 follows the same structure as the sensor model.

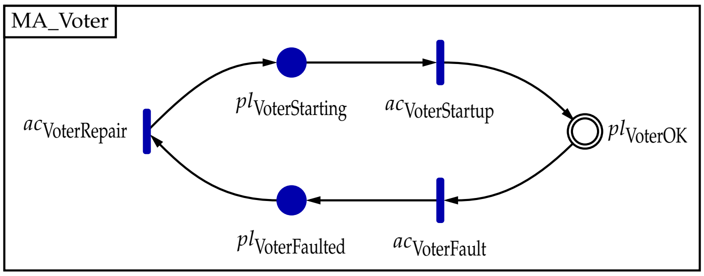


Figure 4.35: HSAN macroactivity model of the overheating sensor.

Figure 4.36 shows the overheating and valve control models.

Table 4.2 shows the bounding table for the fusion places of the Macro Activity Class (MAC) instances in the hierarchical model.

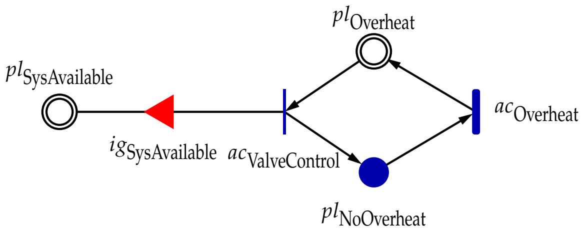


Figure 4.36: HSAN macroactivity model of the control valve and overheating mechanism.

Table 4.2: Fusion of places of the hierarchical model

MAC Instance	**MAC Place**	**Model bound place**
Sensor1	$pl_{SensorOK}$	$pl_{Sensor1OK}$
Sensor2	$pl_{SensorOK}$	$pl_{Sensor2OK}$
Sensor3	$pl_{SensorOK}$	$pl_{Sensor3OK}$
Voter	$pl_{VoterOK}$	$pl_{VoterOK}$
TMRLogic	$pl_{Sens1OK}$	$pl_{Sensor1OK}$
TMRLogic	$pl_{Sens2OK}$	$pl_{Sensor2OK}$
TMRLogic	$pl_{Sens3OK}$	$pl_{Sensor3OK}$
TMRLogic	$pl_{VoterOK}$	$pl_{VoterOK}$
TMRLogic	$pl_{SysAvailable}$	$pl_{SysAvailable}$
Overheating Model	$pl_{SysAvailable}$	$pl_{SysAvailable}$
Overheating Model	$pl_{Overheat}$	$pl_{Overheating}$

Given that Möbius tool does not yet support HSAN models at the time this research was produced, the model was expanded to a flat SAN model for testing purposes. The tool generates an EXtensible Markup Language (XML) file for each model. A C++ tool was developed to parse this XML file and convert each component to equivalent FPGA IP components, and produce a Tool Command Language (TCL) script for Xilinx Vivado® 2018.1 that generates the architecture shown in Figure 4.25.

The activities were parameterized for the specific parameters shown in Table 4.3, synthesized and implemented for a small sized FPGA (XC7Z020-1). In order to simplify its validation, all sensors have been parameterized with the same fault and repair rates, but they could be parameterized with differing values in the same manner. Figure 4.37 shows the conversion of activity $ac_{OverheatingUndetected}$ and the associated gates and places in the SAN model.

Table 4.3: Activity time distributions

Activity	**Distr. Type**	**Parameters**
$ac_{\text{VoterFault}}$	Exp.	λ=2.853 × $10^{-6}\,h^{-1}$
$ac_{\text{VoterRepair}}$	Exp.	λ=1.736 × $10^{-4}\,h^{-1}$
$ac_{\text{SensorFault}}$	Exp.	λ=3.805 × $10^{-6}\,h^{-1}$
$ac_{\text{SensorRepair}}$	Exp.	λ=2.315 × $10^{-4}\,h^{-1}$
$ac_{\text{SensorStartup}}$	Det.	τ=0.1 h
$ac_{\text{VoterStartup}}$	Det.	τ=0 h
ac_{Overheat}	Exp.	λ=3.47 × $10^{-4}\,h^{-1}$

Det. = Deterministic, Exp. = Exponential

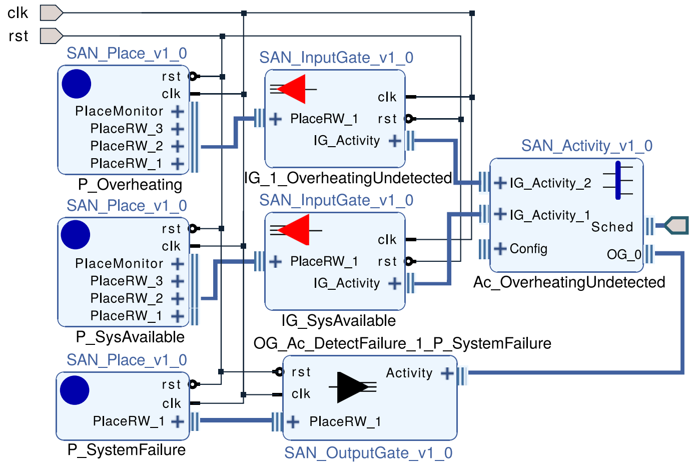


Figure 4.37: FPGA conversion of activity $ac_{OverheatingUndetected}$.

Table 4.4 shows the resulting resource usage of the full SAN simulator implementation and the slacks of the routing, at a clock rate of 100 MHz.

Table 4.4: FPGA resource usage and times

FPGA Device	(XC7Z020-1)	
PL Clock	100 MHz	
Thermal Margin@25.0 °C	39.2 °C	
Resource	**Count**	**(%)**
LUT	11115	20.89
Registers	10768	10.12
DSP	6	2.73
BRAM	1	0.71
Timing analysis results	**Value**	
Worst Negative Slack (WNS)	0.006 ns	
Worst Hold Slack (WHS)	0.031 ns	
Worst Pulse Width Slack (WPWS)	3.750 ns	

The system was capable of fitting 3 full parallel simulators as shown in Figure 4.38, and results were collected with the 2 available CPU cores, following the method described in Section 4.8.

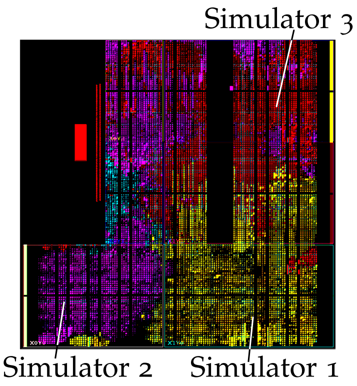


Figure 4.38: Resulting FPGA implementation layout for 3 parallel simulators.

The model was simulated with 100 years of mission time and 3.5×10^6 trials were performed to reach the required 95% confidence interval. FPGA simulations were performed with a 1.76 s event time resolution (2^{-11} h). The captured data size and DMA transfer times depend on the number of rewards and the marking changes of the associated places. Therefore, the same model has been synthesized with varying numbers of reward places in the result collector. In order to evaluate its scalability, the results were also compared with different PDF histogram bin sizes.

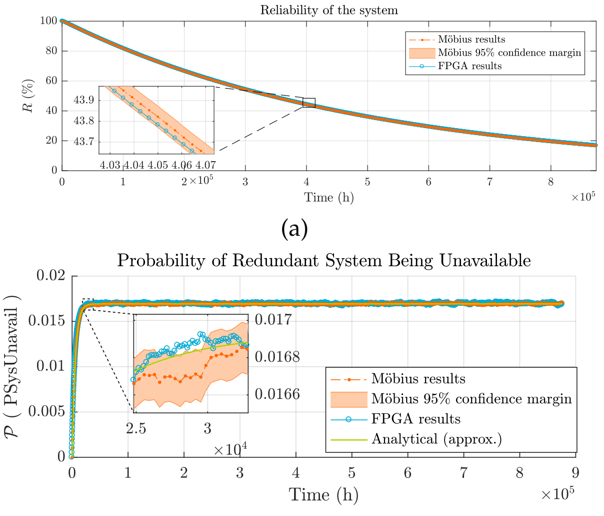
Obtaining an analytical solution for the full system failure PDF for the results validation is complex. The system availability is non-exponentially distributed, even if the failure events are modeled with exponential distributions, and also due to the deterministic sensor startup times. However, an expression for the unavailability can be obtained by neglecting the sensor startup times, approximating the sensor failure PDF with the Kolmogorov differential equations solution for a 2-state Markov chain (Ross, 1996)

$$\mathcal{P}\left(pl_{\text{SensorFaulted}}, t\right) \approx \frac{\lambda_{Fi}}{\lambda_{ri} + \lambda_{ri}} \left(1 - e^{-(\lambda_{Fi}+\lambda_{ri})t}\right) \tag{4.26}$$

where $\mathcal{P}\left(pl_i, t\right)$ denotes the probability that place pl_i is marked with a marking of value greater than zero at time t. λ_{Fi} and λ_{ri} are the i^{th} sensor failure and repair rates respectively.

Results for the reliability of the full system and for the unavailability of the TMR system are compared with SW simulations, and with the analytical solution for the redundant system unavailability, in Figure 4.39a. The confidence margins from the FPGA results have been omitted for clarity. The results were found to converge to the SW and analytical solutions. Therefore, the method was deemed to be valid for reliability analysis purposes.

The efficiency of the simulator architecture was compared against a high-end parallel processing computer system, where SW simulations were run on Möbius, parallelized on 12 cores of an Intel® Xeon® E5-2690 v3 @ 2.60 GHz, with 128 GiB RAM. Table 4.5 compares SW and HW simulation times, including DMA transfer and CPU post-processing times and RAM resource requirements.



(a)

(b)

Figure 4.39: Use case 1 results of (a) full system reliability and (b) redundant system unavailability.

Table 4.5: Resulting simulation timings and memory usage

Monit. places	Output Resolution (h)	Completions	Captured changes (×10^6)	CPU (s)	RAM usage (GiB)	HWsim. time (s)	DMA time/size (s) / (MiB)	Total time (s)	Speed gain	RAM gain
				Simulation time and memory resources						
				SW		**FPGA**				
4	360	1.438×10^9	25.7	6872	39.4	100.46	4.95/196.2	161.00	42.68	205
4	180		25.7	68 500	87.3	100.46	4.95/196.2	296.09	231.34	455
4	90		25.7	n/a(1)	n/a(1)	100.46	4.95/196.2	630.24	n/a(1)	n/a(1)
1	360		6.4	1735	9.90	99.23	1.25/48.95	130.67	13.28	207
1	90		6.4	8267	52.3	99.23	1.25/48.95	159.41	51.85	1094

(1) SW simulation could not be performed with the given resources.
(2) 12 parallel simulations required 2315 s. 10^5 iterations time extrapolated.

The resolution of the output PDF or the bin size of the histograms was found to impact on the performance of the SW simulations, while the FPGA simulation times are not affected, due to the simulation mechanism. The RAM memory usage also grew for the SW simulations and some of the simulations such as the case with four places and 90 h interval could not be run in SW with the given resources.

The FPGA simulations were found to perform 13 to 231 times faster, even if the device only permitted three parallel simulations, given that it was compared on a low resource and low-cost FPGA. The speed gain is mostly achieved by the removal of the CPU and RAM interfacing, despite the relatively low clock frequency achievable on this low-end device. Due to the capture mechanism described in Section 4.7, the memory resources are 207 to 1094 times lower than on the *SW* simulations, reducing the final cost and applicability of the method for on-site and on-line solutions.

4.11 Scalabilily of the architecture

Section 4.10 described the resource usage of multiple parallel scheduler blocks. In comparison, the size of activity and gate blocks is reduced, due to their simplicity, unless complex physical models are integrated. Therefore, their individual resource usage can be considered negligible. However, their usage grows linearly with the model size, and it should be considered for larger models. In order to evaluate the resource usage impact of the full simulator on larger models, the SAN model in Figure 4.40 was synthesized at varying N_{ac} number of activities. The model includes N_{ac} activities that describe a Petri-net like behavior, and activities are completed in sequence. A graph of the resource usage versus the model size is displayed in Figure 4.41. The total LUT usage for 128 activities was found to be 32 371. In comparison, the minimization algorithm corresponds to 18 % of the resources. Therefore, as the model grows, the impact on the resource usage is low.

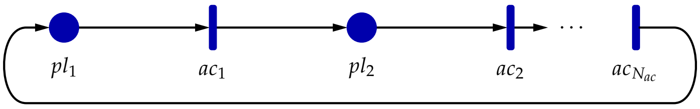


Figure 4.40: Resource scalability test model.

Figure 4.41 shows the corresponding resource usage in the FPGA.

It should also be noticed that the size of the model also impacts on the complexity of the routing for smaller FPGA, and the achievable resulting clock latencies may vary between FPGA devices.

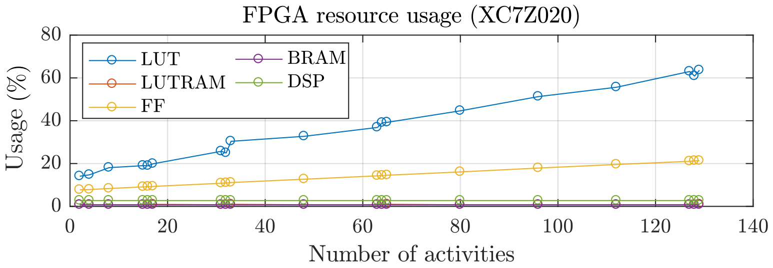


Figure 4.41: Resource scalability test results.

5

Field Programmable Gate Array (FPGA)-BASED ACCELERATED ELECTRO-THERMAL DEGRADATION MODEL OF CABLES

The methods and framework described in Chapter 2 and adapted in Chapter 4 for FPGA simulation permit analyzing the behavior of an electric power distribution grid component subjected to varying grid conditions. A physics-based model of the component can be integrated within a Stochastic Activity Network (SAN) model that represents the behavior of the switching and loading of the grid, where the grid and the component models mutually affect each other. Section 3.3 described a validated approach for the analysis of the degradation of underground cables subjected to varying loads. However, the accelerated simulation of such a degradation model in an FPGA requires adapting the model to a parallelized execution using the limited resources available in the programmable logic of the Hardware (HW).

The framework described in Chapter 4 also requires that the physics-based model acts as an activity in the SAN model, and that the activity must produce a completion time interval for the scheduler as implemented in Chapter 4. This interval corresponds to the Time To Failure (TTF) of the cable, given the instantaneous conditions of the grid.

The physics-based model from Section 3.3 is a time-invariant state-space model where their future behavior depends on the past states or inputs. The inputs are varying in time according to a SAN model of the grid, and new input values are only valid after the completion of each activity and the propagation latency has expired (see Section 4.6). Accordingly, the HW-adapted physical model simulation must be able to pause the simulation while the input data is invalid, then resume the simulation, to run until a given mission time $t_{M,\text{Phys}}$ is reached or a fault condition is met, and to notify this time interval to the scheduler.

In FPGA designs two approaches are followed (Meyer-Baese, 2014, Chapter 2):

i) A single-stage or multi-stage blocks that calculate an output for each given input ;

ii) a pipelined architecture, where the outputs of multiple sequential inputs are being calculated in parallel.

5.1 Pipelined vs Non-Pipelined Physical Models

In order for the physics-based model simulation to be efficient, the latency of the simulation between each SAN simulation activity time generation request until the new mission time is reached must be minimized. However, the electro-thermal degradation models described in Section 3.3 are too complex for their evaluation to be produced in a single clock cycle: (i) Some operations such as divisions can require tens of clock cycles to operate at the required resolutions; (ii) the convolution process means that the state at any given simulation time t cannot be obtained without considering all the past input contributions; and (iii) the degradation functions are non-linear.

Due to not being able to operate in a single clock cycle, a sequential process considering all the past states must be considered. However, the latency of such process would be relatively long, and in the case of underground cables, the thermal responses are slow, and short interval load transitions can have an impact on the thermal analysis. Because of this, a simple multi-stage sequential simulation would not be feasible due to the simulation times and the required FPGA resources.

In a not-pipelined approach, the implementation of state-space models requires a stateful system that considers the past states $\mathbf{X}[n-1]$ and new system inputs $\mathbf{U}[n]$ to produce the new model outputs $\mathbf{Y}[n]$ as shown in Figure 5.1. For complex models the process requires multiple clock cycles and it can be split into several less complex N subprocesses or stages, where each i^{th} stage block S_i has a latency of $\text{ltn}(S_i)$ clock-cycles.

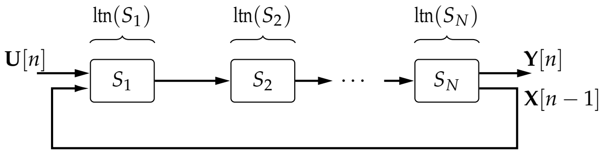


Figure 5.1: Non-pipelining multi-stage architecture.

Notice that in an actual implementation of the architecture the input $\mathbf{U}[n]$ may not be always valid. In the case of the SAN grid model, the inputs are valid only after the SAN simulator completes and propagates the state of the current grid condition. In order to pause the simulation while inputs are invalid, each individual block in Figure 5.1 would require an enablement mechanism. The process of enablement will be covered in the following sections and this mechanism has not been depicted in the figure for readability purposes. The total latency for each sample in this architecture would be

$$\text{ltn}(\{S_1 \dots S_N\}) = \sum_{i=1}^{N} \text{ltn}(S_i) \tag{5.1}$$

In this architecture, inputs are only processed after the previous sample has been fully processed.

On the other hand, in a fully pipelined architecture, each of the segments of the structure are designed to receive sequential input samples with no delay cycles in between, as shown in Figure 5.2

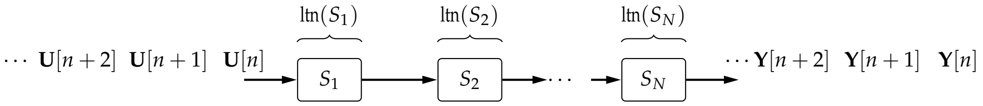


Figure 5.2: Pipelined multi-stage architecture.

In this architecture the latency of the first sample, or the total pipeline latency τ_{pip} is the sum of the latencies of each stage. However, after the first output, the latency of the next output samples, or the sample latency τ_{samp} is equal to the maximum of the latencies of each stage:

$$\tau_{\text{pip}} = \text{ltn}(\{S_1 \dots S_N\}) = \sum_{i=1}^{N} \text{ltn}(S_i) \tag{5.2}$$

$$\tau_{\text{samp}} = \max\left(\{\text{ltn}(S_1), \text{ltn}(S_2), \dots \text{ltn}(S_N)\}\right) \tag{5.3}$$

Therefore, splitting the processes into smaller sub-processes may increase the total latency of the pipeline. However, each successive sample can be put through in very small clock-cycle counts, obtaining a higher throughput. The complexity of this architecture lies in:

i) Each sub-process must be designed to be pipelined which is not always possible;

ii) the pipeline cannot be directly looped back from the output corresponding to an older input sample in stage S_{m+i} to an earlier stage S_m with a newer input sample, given that the output of stage S_{m+i} is not calculated yet by the time S_m is being evaluated.

The latter constraint can be challenging for state-space models with higher order transfer functions, because the output for any sample at time t is to be evaluated with output samples that occurred at $t - \tau$, where the output for the input sample at $t - \tau$ may not yet be fully calculated by the time it is required.

In general, a state-space discrete-time linear time-invariant physical process model is represented as (Ogata, 1994, Chapter 5):

$$\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{B}\mathbf{u}(n) \tag{5.4}$$

$$\mathbf{y}(n) = \mathbf{C}\mathbf{x}(n) + \mathbf{D}\mathbf{u}(n) \tag{5.5}$$

where n is the discrete-time index, $\mathbf{x}(n)$ is the system *state vector*, $\mathbf{u}(n)$ is the system *input vector*, $\mathbf{y}(n)$ is the system *output vector*, and $\mathbf{A}$,$\mathbf{B}$,$\mathbf{C}$ and $\mathbf{D}$ are constant matrices for time-invariant systems.

In a pipelined implementation of Equation (5.4), if the full pipeline to calculate $\mathbf{x}(n)$ has a latency of M clock cycles, the next sample $\mathbf{x}(n+1)$ may require $\mathbf{x}(n)$ to be evaluated M samples earlier, preventing a fully pipelined approach.

However, in the context of a high-reliability equipment SAN model simulation, a pipelined approach is desirable because of the long mission times involved (time between the rare grid events) and the number of samples to be processed between the events. The total simulation time τ_{SIM} of a physics-based model with a mission time $t_{M,\text{Phys}}$ measured in number of samples, for a non-pipelined approach would be:

$$\tau_{\text{SIM}} = t_{M,\text{Phys}} \sum_{i=1}^{N} \text{ltn}(S_i) \tag{5.6}$$

For long mission times $t_{M,\text{Phys}}$, if a pipelined architecture with low sample latency τ_{samp} can be achieved, the simulation time is dramatically reduced even for larger pipeline lengths:

$$\tau_{\text{SIM}} = (t_{M,\text{Phys}} - 1)\tau_{\text{samp}} + \sum_{i=1}^{N} \text{ltn}(S_i) \tag{5.7}$$

In this context, the objective of this chapter is the development of such a pipelined model for the electro-thermal degradation model described in Chapter 3, at the lowest possible sample latency $\tau_{\text{samp}} = 1$ and higher clock-frequencies. The higher clock frequencies are obtained by reducing the complexity of each of the stages, down to the optimal cell sizes of the FPGA architecture (Meyer-Baese, 2014).

5.2 PIPELINE PAUSING AND FLUSHING

The architecture proposed in Chapter 4 requires that the physical model simulator attends to state changes of the grid model. This process is similar to the data hazard conditions commonly found in Central Processing Unit (CPU) pipelines (Omondi, 1999, Chapter 5). The physics-based model of the cable depends on the instantaneous state of the grid, but the grid condition depends on the earlier outputs from the cable model. Therefore, the new inputs to the pipeline, that come from the SAN grid model, are not available until the physical model has not finished the simulation up to the given (partial) mission time $t_{M,\text{Phys}}$.

Therefore, the pipeline cannot proceed calculating and it must stalled until the pipeline is flushed and processes the last valid output, and the SAN simulator provides the new grid inputs.

Accordingly, when an activity affecting the cable condition completes in a SAN model, the pipeline needs to be flushed until the outputs up to the mission time or its fault event are completed and reported to the SAN model. When the new inputs enter the pipeline the simulation can proceed.

In order to implement this stall (pause) and flush mechanism, the segments of the pipeline and the simulation must include an enablement mechanism as shown in Figure 5.3

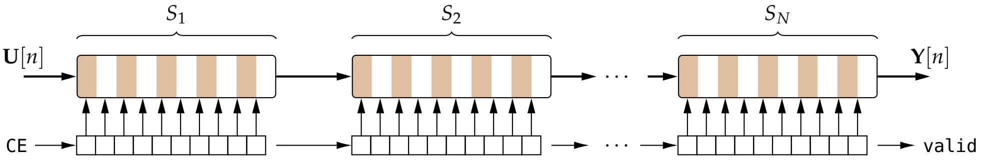


Figure 5.3: Pipelined multi-stage architecture with pipeline pause and flushing mechanism.

A clock enable (`CE`) signal is carried through the pipeline, indicating if the data input to the model at each stage is valid. When '0', the signal indicates each stage that the SAN model completed an activity before this input data was generated, and therefore its value is invalid, causing the stage to pause. The signal is carried through the full pipeline by means of a shift register that enables or disables each cell individually. A `valid` signal is generated at the output that indicates if the last output available from the pipeline is valid or it was generated by non-available inputs.

This process is necessary when the stages in the pipeline are stateful. That is, when the output of an stage depends on the current state of this stage that can depend on the past inputs. Accordingly, accepting any invalid input would cause future outputs to be incorrect. However, often parts of the pipeline are stateless, meaning that even if an output is produced using invalid input, the future outputs would still be valid. In this case, their execution can continue, as far as their output value is ignored. While the energy consumption can increase due to not disabling their execution, it generally reduces routing complexity due to the removal of the `CE` signal, as shown in Figure 5.4:

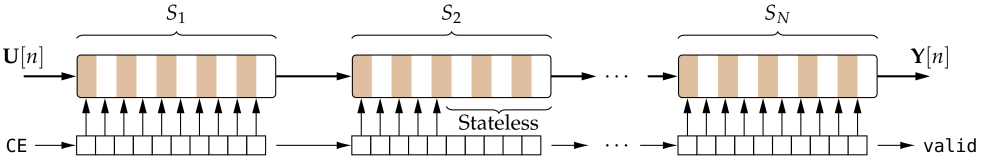


Figure 5.4: Pipelined multi-stage architecture with pipeline pause and flushing mechanism including stateless cells.

5.3 PARALLELIZATION AND SYNCHRONIZATION OF PIPELINE STAGES

When simulating complex processes, the pipelines can be accelerated by multiple parallel lines that evaluate independent system sub-processes. Often, the evaluation of each subprocess has a different pipeline latency. In these cases shift registers can be employed to align sample values corresponding to the required discrete-time instant as shown in Figure 5.5.

Any cell in any of the parallel subprocesses that has larger than 1 clock cycle to operate in these subprocesses forces the parallel subprocesses to align to be able to synchronize for

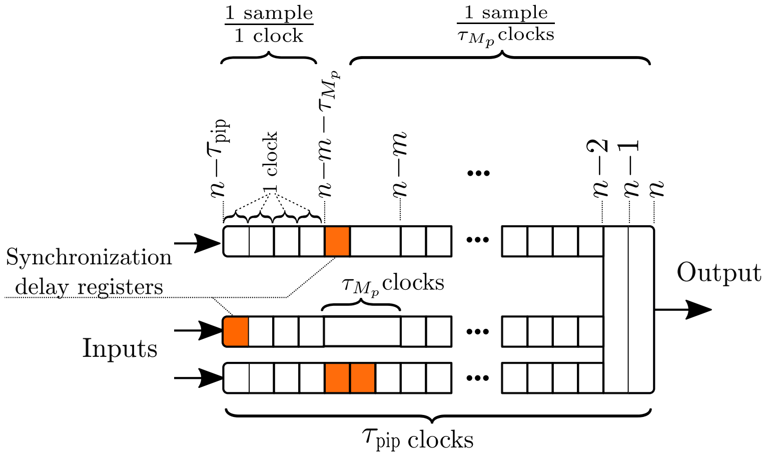


Figure 5.5: Synchronization of pipeline subprocesses with different pipeline latencies.

the next stages, and the cell with the largest cell latency, denoted τ_{M_p} in this figure, defines the pipeline sample latency $\tau_{\text{samp}} = \tau_{M_p}$.

Whenever a data hazard is caused in the pipeline, part of every parallel pipeline must enter a stall mode and perform the flushing of the valid inputs until the last valid output is processed, as described in Section 5.2. This can be implemented by the architecture shown in Figure 5.6, where the `CE` signal is true if the current simulation time t_{SIM} has not reached the mission time $t_{M,\text{Phys}}$ (`CE`$= (t_{M,\text{Phys}} > t_{\text{SIM}})$).

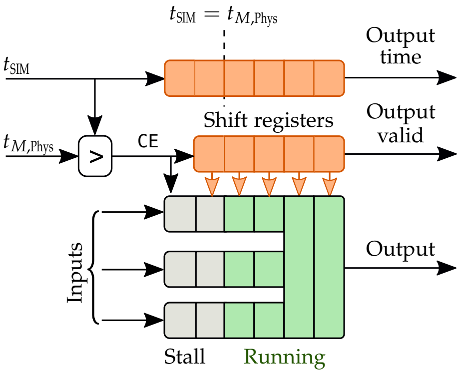


Figure 5.6: Pipeline stall of a physics-based model with parallel pipelines, when mission time $t_{M,\text{Phys}}$ is reached.

Generally physics-based models depend on the simulation time t_{SIM}. Therefore, this value is also carried through the pipeline by a shift-register as shown in Figure 5.6. The output time value can also be used to check if the physics simulation has reached the mission time $t_{M,\text{Phys}}$.

5.4 Cost of Interruption of Pipelines in Physical Models (Switching Time)

Whenever an activity in the SAN model completes, the new states (markings) affecting the physical model simulation can be altered and the pipeline enters a stall and flush mode. After this time, the pipeline must be buffered again, and the pipeline output will be invalid until the new data goes through the full pipeline. Therefore, each activity completion causes a delay equal to the pipeline latency τ_{pip}.

This delay can be prevented if the list of activities that can affect the physical model simulation can be identified. In that case, the calculation of in $t_{M,\text{Phys}}$ in Equation (2.8) can be limited to that activity set. The process is generally simple and possible to automatize, by identifying all output and input gate functions that alter the markings and identifying the associated activities. However, the implementation of this mechanism would not be a general solution, as it requires the customization of the scheduler to each specific SAN model, or the addition of an activity property as indicator of this dependency relationship with the physics-based model activity ac_{Phys}.

The total simulation time increase due to the pipeline flushing can be evaluated if the expected number of completions for each activity in a simulation is known. However, this can be a challenging task for most grid models, given that the completion of activities in a SAN model is dependent on the enablement by other activities in the model, that is, activity completions cannot be analyzed independently of each other.

5.5 An Electro-thermal degradation Model Optimized for FPGA

From the previous sections, it can be concluded that:

i) A pipelined architecture for the simulation of the electro-thermal degradation models of the cable is desirable;

ii) However, a mechanism is necessary for its stall and flushing;

iii) And the maximum block latency τ_{M_p} should ideally be 1 clock-cycle, which would lead to a τ_{samp}=1 clock-cycles.

As described in Section 3.3.1, the thermal model of a cable must cover the following effects:

i) Main conductor losses;

ii) Induced sheath and armor losses;

iii) Self-heating process of the cable;

iv) Effect of the underground burying of the cable;

v) Mutual-heating caused by adjacent cables;

vi) and the varying resistivity of the conductors with temperature.

The self-heating process was described by a transfer function in Equation (3.30) that can be adapted for a discrete-time FPGA implementation by a z-transform conversion of the transfer function. It is desired that the implmentation for this z-transform is pipelined with a sample latency of $\tau_{\text{samp}} = 1$ clock-cycle. The process is described in Section 5.5.2. However, the models described by the standard (IEC, 2008) for the effect of burying the cable and the mutual heating processes between cables are not described as a transfer function. Those models must be adapted for an adequate discrete representation that enables its pipelined implementation in an FPGA. These implementations are covered in Section 5.5.1.

5.5.1 *Soil effect and mutual heating processes*

The mutual heating process and the effect of the cable being buried underground described by Equation (3.36) and Equation (3.35) are complex to be implemented in an FPGA due to the nonlinear exponential integral function. In addition, obtaining the thermal response to multiple transients requires to implement the superposition of effects by implementing the convolution in Equation (3.39). This convolution covers the superposition of thousands of samples that describe the slow thermal response, and requires large register resources.

Convolutions in FPGA can be implemented as a form of Infinite Impulse Response (IIR) or Finite Impulse Response (FIR) filters (Meyer-Baese, 2014, Chapters 3 and 4). However, the former method requires a discrete transfer function to be obtained, which not always exists. And the FIR method offers a finite time limitation for the impulse response, and can require a large number of weight multipliers for long term responses.

The impulse response of both the mutual-heating and soil effect processes can be obtained from the step response equations. For a constant load W applied at $t = 0$, and considering that the system is causal, the convolution equation Equation (3.39) can be transformed into

$$\theta_{Wv}(t) = W \int_0^{\infty} \theta_\delta(\tau) \mathrm{d}\tau \tag{5.8}$$

where $\theta_{Wv}(t)$ is the response to of the thermal process to a constant W dissipated power increase at instant $t = 0$. Therefore, the impulse response of any of the thermal processes can be obtained from the step response by

$$\theta_\delta(t) = \frac{\mathrm{d}}{\mathrm{d}t} \frac{\theta_{Wv}(t)}{W} \tag{5.9}$$

From Equation (A.13) the first derivative of $\mathrm{E}_1(x)$ can be expressed as

$$\frac{\mathrm{dE}_1(x)}{\mathrm{d}x} = -E_0(x) = -\frac{e^{-x}}{x} \tag{5.10}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}x}\mathrm{E}_1(g(x)) = -\frac{e^{-g(x)}}{g(x)} \frac{\mathrm{d}}{\mathrm{d}x} g(x) \tag{5.11}$$

Therefore, the respective impulse responses to the mutual heating process and soil effect $\theta_{\delta pk}$ and $\theta_{\delta e}$ can be obtained from the step responses Equation (3.36) and Equation (3.35) as

$$\theta_{\delta pk} = \frac{\mathrm{d}}{\mathrm{d}t} \frac{\theta_{pk}(t)}{W_t} = \frac{\mathrm{d}}{\mathrm{d}t} \frac{W_t \frac{\varrho_\mathrm{s}}{4\pi} \left[\mathrm{E}_1\left(\frac{d_{pk}^2}{4\delta_\mathrm{s} t}\right) - \mathrm{E}_1\left(\frac{d'^2_{pk}}{4\delta_\mathrm{s} t}\right) \right]}{W_t} \tag{5.12}$$

$$= \frac{\varrho_\mathrm{s}}{4\pi t} \left(\mathrm{e}^{-\frac{d_{pk}^2}{4\delta_\mathrm{s} t}} - \mathrm{e}^{-\frac{d'^2_{pk}}{4\delta_\mathrm{s} t}} \right) \tag{5.13}$$

$$\theta_{\delta e} = \frac{\mathrm{d}}{\mathrm{d}t} \frac{\theta_e(t)}{W_t} = \frac{\mathrm{d}}{\mathrm{d}t} \frac{W_t \frac{\varrho_\mathrm{s}}{4\pi} \left[\mathrm{E}_1\left(\frac{D_e^2}{16\delta_\mathrm{s} t}\right) - \mathrm{E}_1\left(\frac{L_d^2}{\delta_\mathrm{s} t}\right) \right]}{W_t} \tag{5.14}$$

$$= \frac{\varrho_\mathrm{s}}{4\pi t} \left(\mathrm{e}^{-\frac{D_e^2}{16\delta_\mathrm{s} t}} - \mathrm{e}^{-\frac{L_d^2}{\delta_\mathrm{s} t}} \right) \tag{5.15}$$

Figure 5.7 shows the thermal step and impulse responses of the mutual-heating and soil effect processes that represent Equations (3.35), (3.36), (5.13) and (5.15). The mutual-heating responses have been multiplied by 10 in this figure, due to their smaller scale in comparison to the soil effect.

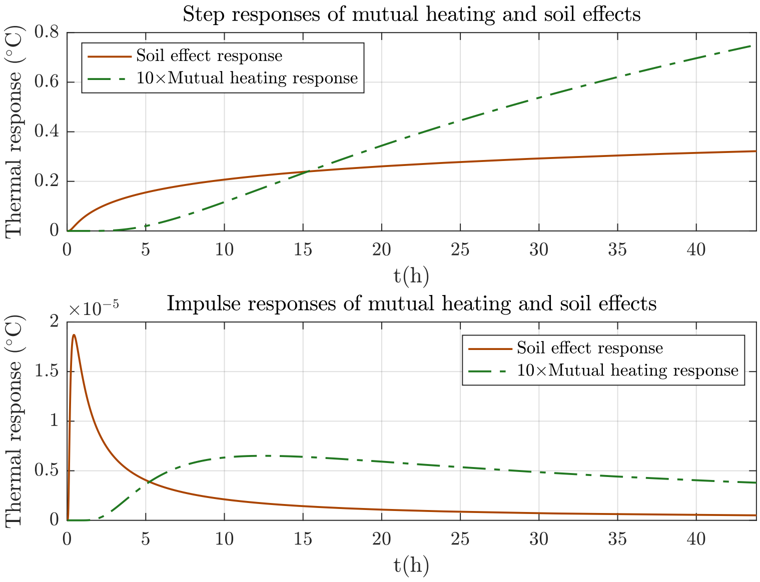


Figure 5.7: Step and impulse thermal responses of the mutual heating and soil effect processes of Cable A in Appendix A.1.

Following the IIR filter approach, when obtaining a transfer function for both thermal heating processes the Laplace transform of a time response in the form of $e^{-\frac{\kappa}{t}}$ results in (Abramowitz and Stegun, 1965, Chapter 29)

$$\mathcal{L}\{e^{-\frac{\kappa}{t}}\} = \sqrt{\frac{\kappa}{4s}}\,\mathrm{K}_1\left(\sqrt{\frac{\kappa}{4s}}\right) \tag{5.16}$$

where $\mathrm{K}_1(\cdot)$ is the modified Bessel function

$$\mathrm{K}_\alpha(x) = \frac{\pi}{2}\lim_{a\to\alpha}\frac{I_{-\alpha}(x) - I_\alpha(x)}{\sin a\pi} \tag{5.17}$$

with

$$I_\alpha(x) = \sum_{m=0}^{\infty}\frac{1}{m!\,\Gamma(m+\alpha+1)}\left(\frac{x}{2}\right)^{2m+\alpha} \tag{5.18}$$

where $\Gamma(\cdot)$ is the gamma function.

The non-linear Laplace transform described by Equation (5.16) is inconvenient for its analysis and a discrete-time implementation, as no equivalent z-transform is known for it. Certain authors (Diaz-Aguilo and Leon, 2015) have approached the solution by approximating the model with an equivalent segmented RC ladder circuit. This model is convenient for computer simulation, but it can lead to varying errors depending on the soil type, and the complete circuit transform can result in higher order transfer functions with increased latencies.

Lookup Table (LUT) and polynomial fitting solutions to represent the impulse responses have also been considered, because they are convenient for FPGA solutions. However, the impulse responses of the thermal processes are not monotonically increasing, their derivatives are non-linear, as shown in Figure 5.7 and segmenting them for an accurate solution can be complex and require more resources. Consequently, the proposed solution is based on employing the convolution based on the step responses described by Equation (3.35) and Equation (3.36).

5.5.1.1 *Discrete step-response based convolution for the thermal model*

The process described by Equation (3.1) is a Linear Time-Invariant (LTI) system. Therefore, its response can be described by the convolution of the impulse response with the dissipated power input as in Equation (3.39), where the input power is considered as the integral sum of $W(\tau)\delta(t-\tau)$ weighed Dirac pulses.

However, employing the principle of superposition by decomposing the input power into step signals as follows, the system thermal response to a varying dissipated power $W(t)$ that is initially in steady state can be approximated by the superposition of multiple step responses separated by a constant T time interval. Defining $v(t)$ the unit step signal

$$v(t) = \begin{cases} 0 & t < 0, \\ 1 & t \geq 0 \end{cases} \tag{5.19}$$

the input power signal can be decomposed as in Figure 5.8

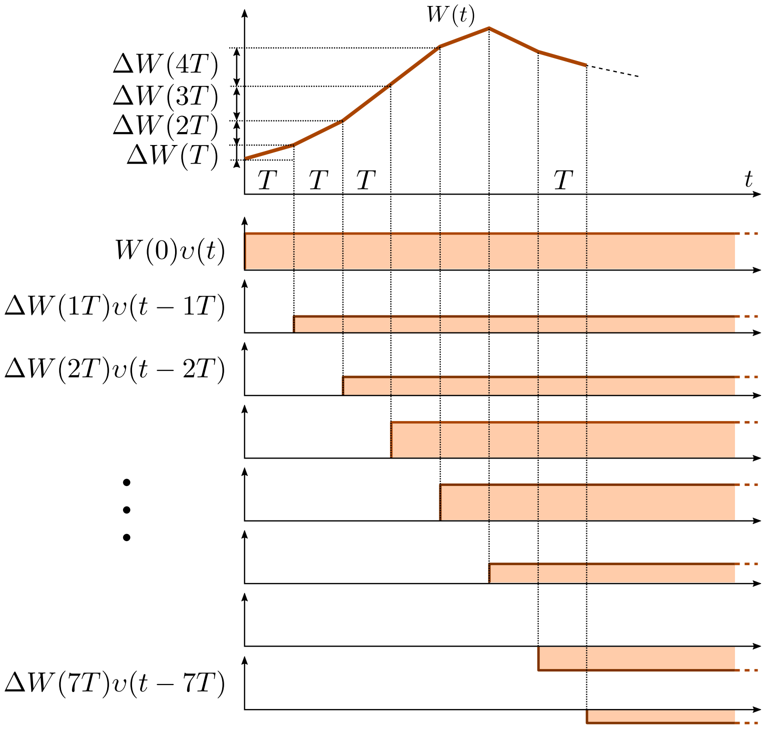


Figure 5.8: Decomposition of the dissipated power signal $W(t)$ into weighed step signals.

By this procedure, the dissipated power signal $W(t)$ value can be defined as the signal in the limit where the interval T tends to 0

$$\begin{aligned} W(t) &= \lim_{T\to 0} \sum_{n=0}^{\infty} \left(W(nT) - W((n-1)T)\right) v(t-nT) \\ &= \lim_{T\to 0} \sum_{n=0}^{\infty} \Delta W(nT) v(t-nT) \\ &= \int_0^t \frac{\mathrm{d}W(\tau)}{\mathrm{d}\tau} v(t-\tau) \mathrm{d}\tau \end{aligned} \tag{5.20}$$

and defining $\theta_{vx}(t)$ as the response of a LTI thermal model x to a unit step power input, the thermal response $\theta_x(t)$ of the thermal model x to any dissipated power transient $W(t)$ can be calculated by the principle of superposition as

$$\begin{aligned}\theta_x(t) &= \lim_{T\to 0}\sum_{n=0}^{\infty}\Delta W(nT)\theta_{vx}(t-nT)\\ &= \int_0^t \frac{\mathrm{d}W(\tau)}{\mathrm{d}\tau}\theta_{vx}(t-\tau)\mathrm{d}\tau\\ &= \frac{\mathrm{d}W(t)}{\mathrm{d}t} * \theta_{vx}(t)\end{aligned} \tag{5.21}$$

Equally, a discrete version of the convolution can be employed for a FPGA analysis of the thermal process, where the sampled thermal response $\theta_x[n]$ is sampled at τ_s time interval as

$$\theta_x[n] = \theta_x(n\tau_s) \tag{5.22}$$

where n denotes the discrete-time index.

In a discrete convolution calculation beginning at time $t = 0$ and being thermal processes causal, the thermal process is similarly evaluated by superposition of $\theta_{\delta x}[n]$ impulse responses of the system to discrete $\delta[n]$ *unit impulse* dissipated power input signals as (Oppenheim et al., 1996)

$$\theta_x[n] = W[n] * \theta_{\delta x}[n] = \sum_{i=0}^{\infty} W[n-i]\theta_{\delta x}[i] = \sum_{i=-\infty}^{n} W[i]\theta_{\delta x}[n-i] \tag{5.23}$$

or if the input power is 0 for $t < 0$

$$\theta_x[n] = \sum_{i=0}^{n} W[i]\theta_{\delta x}[n-i] \tag{5.24}$$

which can be represented in function of unit step responses $\theta_{vx}[n]$ as

$$W[n] = \sum_{j=0}^{n}\Delta W[j]v[n-j] \tag{5.25}$$

$$\begin{aligned}\theta_x[n] &= \sum_{i=0}^{n} W[i]\theta_{\delta x}[n-i] = \sum_{i=0}^{n} W[n-i]\theta_{\delta x}[i]\\ &= \sum_{i=0}^{n}\sum_{j=0}^{n-i}\Delta W[j]v[n-i-j]\theta_{\delta x}[i]\end{aligned} \tag{5.26}$$

as $v[n] = 0, \forall n < 0$, the $(n-i)$ upper limit of the sum can be replaced by (n) resulting

$$\begin{aligned}\theta_x[n] &= \sum_{i=0}^{n}\sum_{j=0}^{n}\Delta W[j]v[n-i-j]\theta_{\delta x}[i] = \sum_{j=0}^{n}\sum_{i=0}^{n}\Delta W[j]v[n-i-j]\theta_{\delta x}[i] =\\ &= \sum_{j=0}^{n}\Delta W[j]\sum_{i=0}^{n} v[n-i-j]\theta_{\delta x}[i] = \sum_{j=0}^{n}\Delta W[j]\theta_{vx}[n-i]\\ &= \Delta W[n] * \theta_{vx}[n]\end{aligned} \tag{5.27}$$

where $\Delta W[n]$ is the increment in power dissipation at discrete-time instant n, and $\theta_{vx}(t)$ is the step response of the thermal process. Therefore, the response of any thermal model whose unit step response is known, can be obtained by convolving the thermal dissipation increment signal with the unit step response of the thermal model.

Employing the step response offers advantages over the convolution based on the discrete impulse response:

i) The time constants governing the step response of the thermal models are much lower than the time constants required for their impulse response. Therefore, the response of a system can be obtained by a lower sampling rate when employing the convolution based on the step response;

ii) The step responses are monotonically increasing functions. Therefore, their characterization employing numerical and polynomial fitting can be simplified;

iii) And the step responses commonly employed in the thermal analysis of the cable can be directly applied, and the result to step inputs will always be as accurate as the resolution used in the discretization process, as opposed to the impulse response where errors can be cumulative in such situation.

However, as it will be discussed in Section 5.5.1.2, the implementation of the convolution function in Equation (5.27) poses several issues:

i) The step response is highly non-linear and cannot be approximated by reduced order polynomial transfer functions;

ii) The time required for the step response to reach a value that is within the limits for the signal to be considered in steady state is too long for an FPGA HW based storage;

iii) The routing required for the implementation of the convolution for a reduced latency can be challenging.

5.5.1.2 *Characterization of the thermal step responses*

The mutual-heating and soil effect thermal processes described by Equation (3.36) and Equation (3.35) and their superposition can be simulated by the convolution based on the step responses as described in Equation (5.27). The sampling interval of the step response defines the time resolution of the thermal simulation.

However, the time resolution of the thermal behavior and the degradation of the cable also defines the time resolution of the statistical properties of the cable within the system model. The statistical system model where the degradation model is integrated requires considering fast evolving operational events that govern the behavior of the system, together with fault events that describe the different mechanisms that may lead the system to a failed state. The cable insulator polymer degradation process evolves between these events and it can alter the behavior of the full grid model as shown in Figure 5.9. Therefore, the degradation level needs to be re-evaluated before any event is executed, that is, before any activity is completed. It also requires that the thermal model must be able to produce temperature/degradation samples for the fastest of the sampling rates required by the grid model. Therefore, the sampling time τ_s of the step responses must be equal or lower than this minimal value, or considering that the temperature and degradation level of the cable remains constant until the next physical model sample. In the latter case, the degradation model would also assume a constant temperature within this interval.

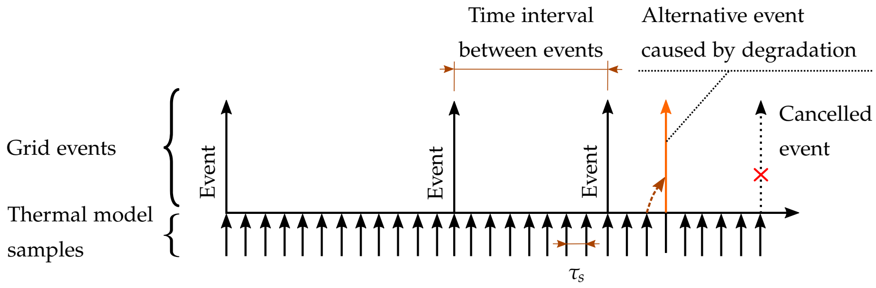


Figure 5.9: Thermal simulation timing within the system model and its simulation.

The convolution in Equation (5.27) can be described by a direct FIR implementation as:

$$y[n] = \sum_{i=0}^{N} \omega_i \cdot x[n-i] \tag{5.28}$$

where N is the filter length, ω_i is the i^{th} filter weight of the FIR filter and $x[n]$ is the input signal, corresponding to the $\Delta W[n]$ samples. In the case of the mutual-heating model, the weights correspond to the discrete-time step response of the mutual-heating process $\theta_{pk}[n]$

$$\omega_i = \theta_{pk}(i\tau_s) = \theta_{pk}[i] \tag{5.29}$$

and for the temperature increase due to the burying in the soil, the weights correspond to the discrete-time step response of the soil effect $\theta_e[n]$

$$\omega_i = \theta_e(i\tau_s) = \theta_e[i] \tag{5.30}$$

In both cases, the input to the system is the total dissipated power density increase $\Delta W[n]$ in the cable that causes the heating process.

Unfortunately, both the soil effect and mutual-heating processes require months before the temperature reaches the steady-state conditions. As it can be observed in Figure 5.10, any constant dissipated power applied to the adjacent cables of Cable A setup in Figure A.1 of Appendix A.1 causes a temperature rise on the central cable that requires more than 20×10^3 h before the temperature reaches within a 1% difference from the steady state condition, and 96×10^3 h for a 0.2% temperature difference. The soil effect in this case reaches its steady state earlier, with 52×10^3 h to reach within a 0.2% difference with the steady state condition. This implies that the length of the FIR filters necessary to estimate the convolution accurately would be unfeasible with current FPGA resources.

The convolution implies evaluating the sum of the responses to multiple power transients by applying the superposition as shown in Figure 5.11. The filter length must be capable to produce the full thermal response up to the end of the simulation. However, the response of a FIR filter is limited by its length, and any output after that point will be null, unless the response after this point is approximated by another architecture as a constant or recurrent process that may represent an Infinite Impulse Response (IIR). And even when approximating the response to a steady state condition after certain samples is possible, the length of the filter required for a given accuracy may be to large to be implemented in an FPGA do to the slow response (see Figure 5.10), and the simulation can in certain cases cover several years or even decades.

It is known that a resolution of a few hours for the thermal transients is required for obtaining an accurate estimation of the degradation of underground cables, due to their

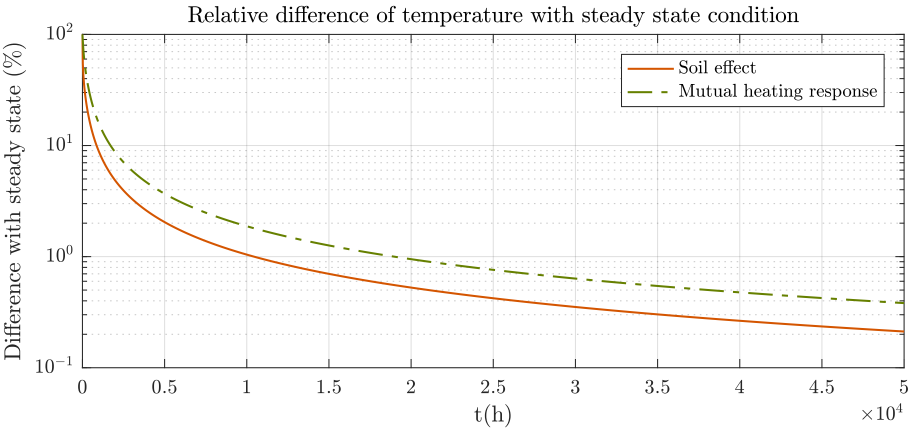


Figure 5.10: Difference between the temperature rises due to the soil effect and the mutual heating with the steady state temperature condition.

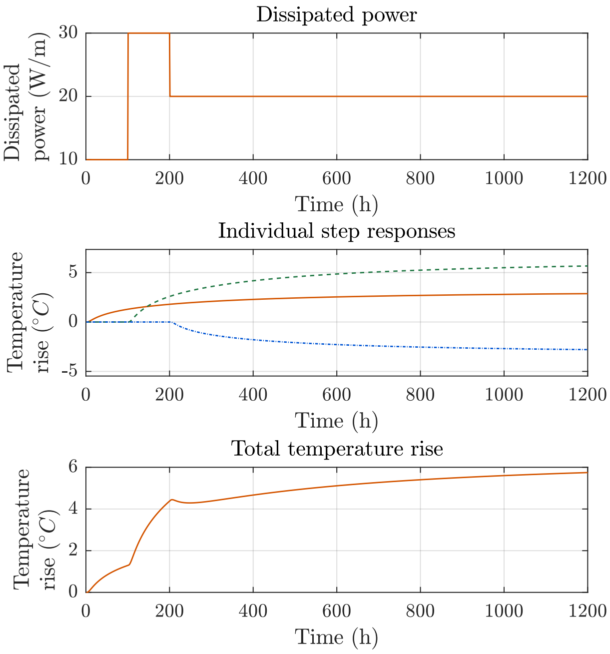


Figure 5.11: Superposition of thermal responses.

temperature dynamics (Mazzanti, 2009). Accordingly, a $\tau_s = 1\,\text{h}$ time resolution for the thermal evolution has been adopted throughout this thesis. However, the implementation of a FIR filter with more than 100×10^3 samples to reach its steady state condition or a longer filter with covering samples up to the end of the simulation was found to be unfeasible due to the resource limitations within current FPGA.

In this context, it is necessary to characterize the thermal processes and produce an alternative form of the convolutions, in order for the FPGA implementation to be feasible in current FPGA.

The derivative of the thermal step responses corresponds to the impulse response of the thermal responses (see Equations (5.13) and (5.15) and Figure 5.7). This response has an increasing derivative in the initial hours, and the derivative growth reaches its minimum after certain number of hours where the derivative of the step response begins decreasing. This time value can be evaluated by the second derivative of the step responses as

$$
\begin{aligned}
\frac{\mathrm{d}^2}{\mathrm{d}t^2}\frac{\theta_{pk}(t)}{W_t} &= \frac{\mathrm{d}}{\mathrm{d}t}\theta_{\delta\theta_{pk}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\varrho_\mathrm{s}}{4\pi t}\left(\mathrm{e}^{-\frac{d_{pk}^2}{4\delta_\mathrm{s}t}} - \mathrm{e}^{-\frac{d'^2_{pk}}{4\delta_\mathrm{s}t}}\right) \\
&= \frac{\varrho_\mathrm{s}}{4\pi t^2}\left[\mathrm{e}^{-\frac{d'^2_{pk}}{4\delta_\mathrm{s}t}}\left(1-\frac{d'^2_{pk}}{4\delta_\mathrm{s}t}\right) - \mathrm{e}^{-\frac{d^2_{pk}}{4\delta_\mathrm{s}t}}\left(1-\frac{d^2_{pk}}{4\delta_\mathrm{s}t}\right)\right]
\end{aligned}
\tag{5.31}
$$

$$
\begin{aligned}
\frac{\mathrm{d}^2}{\mathrm{d}t^2}\frac{\theta_{e}(t)}{W_t} &= \frac{\mathrm{d}}{\mathrm{d}t}\theta_{\delta\theta_{e}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\varrho_\mathrm{s}}{4\pi t}\left(\mathrm{e}^{-\frac{D_e^2}{16\delta_\mathrm{s}t}} - \mathrm{e}^{-\frac{L_d^2}{\delta_\mathrm{s}t}}\right) \\
&= \frac{\varrho_\mathrm{s}}{4\pi t^2}\left[\mathrm{e}^{-\frac{L_d^2}{\delta_\mathrm{s}t}}\left(1-\frac{L_d^2}{\delta_\mathrm{s}t}\right) - \mathrm{e}^{-\frac{D_e^2}{16\delta_\mathrm{s}t}}\left(1-\frac{D_e^2}{16\delta_\mathrm{s}t}\right)\right]
\end{aligned}
\tag{5.32}
$$

The maximum derivate of the step reponses therefore occurs when the second derivate functions equal to zero, that is

$$
\arg\max \frac{\mathrm{d}}{\mathrm{d}t}\theta_{pk}(t) = \frac{-\frac{d_{pk}^2}{4\delta_\mathrm{s}} - \frac{-\frac{d'^2_{pk}}{4\delta_\mathrm{s}}}{\mathrm{e}^{\left(-\frac{d^2_{pk}}{4\delta_\mathrm{s}}+\frac{d'^2_{pk}}{4\delta_\mathrm{s}}\right)}}}{\mathrm{e}^{\left(-\frac{d'^2_{pk}}{4\delta_\mathrm{s}}+\frac{d^2_{pk}}{4\delta_\mathrm{s}}\right)}-1} \approx \frac{d^2_{pk}}{4\delta_\mathrm{s}}
\tag{5.33}
$$

$$
\arg\max \frac{\mathrm{d}}{\mathrm{d}t}\theta_{e}(t) = \frac{-\frac{D_e^2}{16\delta_\mathrm{s}} - \frac{-\frac{L_d^2}{\delta_\mathrm{s}}}{\mathrm{e}^{\left(-\frac{D^2_{e}}{16\delta_\mathrm{s}}+\frac{L^2_{d}}{\delta_\mathrm{s}}\right)}}}{\mathrm{e}^{\left(-\frac{L^2_{d}}{\delta_\mathrm{s}}+\frac{D^2_{e}}{16\delta_\mathrm{s}}\right)}-1} \approx \frac{D^2_{e}}{16\delta_\mathrm{s}}
\tag{5.34}
$$

It can be observed that this inflection point occurs at different time values for the mutual-heating and soil effect thermal responses. These two highest temperature increase points occur at relatively early times after the power increase has been injected to the cable, although the time value depends on the dimensions of the cable setup and the thermal diffusivity (see Section 3.3.4). Both soil effect and mutual-heating effects are monotonically increasing processes that have their maximum derivate in very short times, and progressively reduce the rate after this time instant, as shown in Figure 5.12, that corresponds to the thermal response of Cable A setup in Appendix A.1.

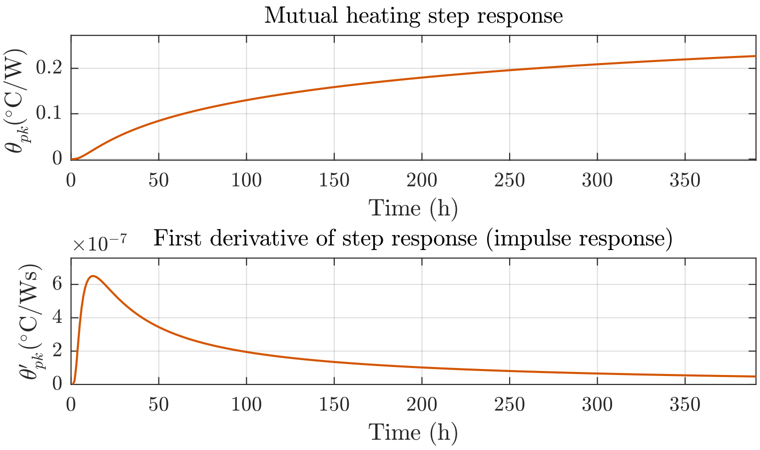


Figure 5.12: Derivative functions of the step response of the mutual-heating process.

Plotting the curves in a semilogarithmic axes, it can be observed in Figure 5.13 that an inflection occurs in the step response, versus $\log(t)$. It can be appreciated that the inflection point in this case occurs at $t = 192\,\mathrm{h}$.

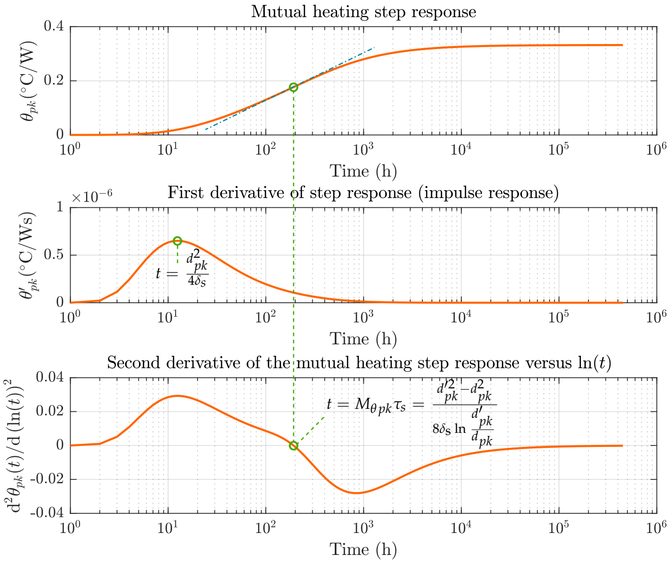


Figure 5.13: Inflection point of step response with respect to $\log(t)$.

This inflection point corresponds to $\frac{\mathrm{d}^2\theta(t)}{\mathrm{d}(\ln(t))^2} = 0$, being

$$\frac{\mathrm{d}^2\theta(t)}{\mathrm{d}\left(\ln(t)\right)^2} = t^2\frac{\mathrm{d}^2\theta(t)}{\mathrm{d}t^2} + t\frac{\mathrm{d}\theta(t)}{\mathrm{d}t} \tag{5.35}$$

where the first and second derivatives are given by Equations (5.13) and (5.15), and Equations (5.31) and (5.32) respectively.

The inflection point for the mutual-heating process occurs after $M_{\theta pk}$ samples at time

$$t = M_{\theta pk}\tau_s = \frac{d'^2_{pk} - d^2_{pk}}{8\delta_s \ln \frac{d'_{pk}}{d_{pk}}} \tag{5.36}$$

and for the soil effect process after $M_{\theta e}$ samples at time

$$t = M_{\theta e}\tau_s = \frac{L_d^2 - \frac{D_e^2}{16}}{2\delta_s \ln \frac{4L_d}{D_e}} \tag{5.37}$$

At this time instant, the first derivative of the function with respect to the logarithm of the time approximates a maximum value, and decreases rapidly afterwards. At the inflection time instant, the curve tends to a linear derivative point versus $\log(t)$. This means that the thermal response grows logarithmically, or the rate of the derivative function decreases geometrically. From this point on, the growth of the thermal response versus the logarithm of time is slower than linear. Which means the thermal response is decelerating even faster at growing exponent values.

Similarly, Figure 5.14 shows the second derivative of the soil effect with respect to the logarithm of the time for the Cable A setup. It can be observed that the inflection point for this thermal process occurs at $t = 106\,\mathrm{h}$.

Therefore, both the step responses of the soil effect and mutal-heating processes exhibit two regions:

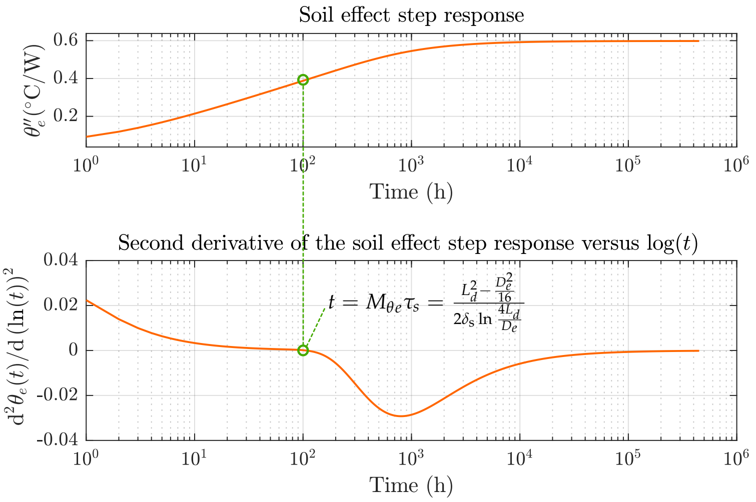


Figure 5.14: Inflection point of the step response of the soil effect with respect to log(t).

i) A growing derivative of the curve that occurs during the early hours after the power increase is applied, up to the time instants given by Equations (5.33) and (5.34);

ii) A geometrically decreasing growing-rate after M_θ samples given by Equations (5.36) and (5.37).

In this context, the geometrically decreasing derivative of the thermal responses after the M_θ samples can exploited to reduce the required filter sizes in Equation (5.28) as it will be described in Section 5.5.1.3.

5.5.1.3 *Segmentation of the thermal step responses*

When the coefficients of a FIR filter remain constant within a given accuracy margin, they can be approximated to this constant value that replaces them. Accordingly, the geometrically decreasing rate of the derivatives of the thermal processes described in Section 5.5.1.2 can be exploited by implementing a variable sampling rate of the step responses. The sampling rate can be increased as the derivative of the step response decreases. This decrease in the derivative function occurs after a given point in each step response M_θ, that can be calculated from Equations (5.36) and (5.37).

This approach produces a convolution with a varying sample rate by dividing the convolution into several subprocesses of varying response length. In general, the discrete convolution sum based on the step response of any thermal process as in Equation (5.27) can be split into N sub-processes or segments as:

$$\begin{aligned}
\theta_x[n] =& \Delta W[n] * \theta_v[n] = \sum_{i=0}^{n} \theta_v[i]\Delta W[n-i] \\
=& \sum_{i=P_1=0}^{P_2-1} \theta_v[i]\Delta W[n-i] + \sum_{i=P_2}^{P_3-1} \theta_v[i]\Delta W[n-i] + \cdots + \sum_{i=P_N}^{n} \theta_v[i]\Delta W[n-i] \\
=& \sum_{l=0}^{N} \sum_{i=P_l}^{P_l+L_l-1} \theta_v[i]\Delta W[n-i] \\
=& \sum_{l=1}^{N} \sum_{j=0}^{L_l-1} \theta_v\left[j+P_l\right]\Delta W\left[n-j-P_l\right] \\
=& \sum_{l=1}^{N} \sum_{j=0}^{L_l-1} \theta_v\left[j+P_l\right]\left(\delta\left[n-P_l\right] * \Delta W[n-j]\right)
\end{aligned}$$

$$= \sum_{l=0}^{N} \delta\left[n - P_l\right] * \Delta W[n] * \theta_{v,P_l,L_l}[n] \tag{5.38}$$

being L_l the length of the l^{th} segment and $P_l = \sum_{i=1}^{l-1} L_i$ is the discrete time index of the $\theta_v[n]$ thermal response sample where the filter is sectioned, and each segment filter $\theta_{v,P_l,L_l}[n]$ is defined as

$$\theta_{v,P,L_l}[n] = \begin{cases} 0 & n < 0, \\ \theta_v[n+P] & 0 \leq n < L_l, \\ 0 & n \geq L_l \end{cases} \tag{5.39}$$

According to Equation (5.38) the step response of any thermal process can be represented as the sum of multiple FIR filters where the input of each filter has been delayed P_l samples, and the filter coefficients of the l^{th} filter are the values of $\theta_v[n]$ thermal unit step response at offset P_l with length L_l. This architecture is shown in Figure 5.15.

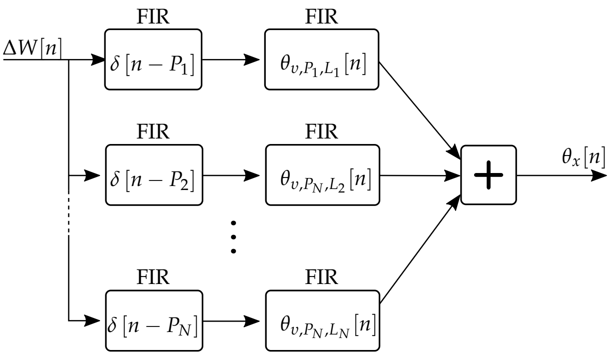


Figure 5.15: Segmented convolution architecture.

Notice the last (N^{th}) filter segment needs to perform the convolution sum up to the n^{th} sample of the step response, where n is the time index of the simulation mission time $t_{M,\text{Phys}}$, or the time index where the thermal response value needs to be obtained. Due to the time length of the thermal analysis in high-reliability equipment, this filter segment would ideally need to be implemented as an IIR filter as described in Section 5.5.3 .

5.5.1.4 *Latency and routing issues of the FIR implementation*

A pipelined approach for a FIR filter implementation can be achieved with a direct FIR filter implementation as shown in Figure 5.16

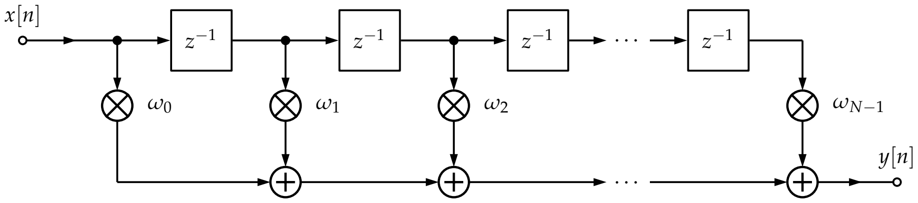


Figure 5.16: Direct form implementation of a FIR filter.

However, the structure in Figure 5.16 has a pipeline with latency $\tau_{\text{pip}} = N$ clocks, being N the length of the filter. Considering the length of the thermal responses, the impact on the latency of the ac_{Phys} would be unacceptable for practical use. A parallel approach such as the one shown in Figure 5.17 partially reduces the latency but causes routing issues that are complex to be solved on larger filter sizes.

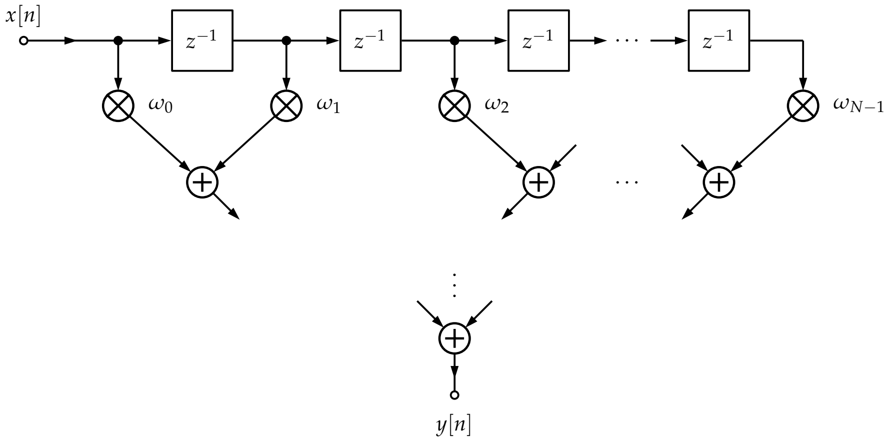


Figure 5.17: A parallel implementation of the FIR filter adder causes complex routing issues.

An alternative is the utilization of a transposed filter structure shown in Figure 5.18 that produces the first sample with a pipeline latency of $\tau_{\text{pip}} = 2$ clocks, and saves a register for each filter implemented, in comparison to the direct filter implementation (Meyer-Baese, 2014, Chapter 3).

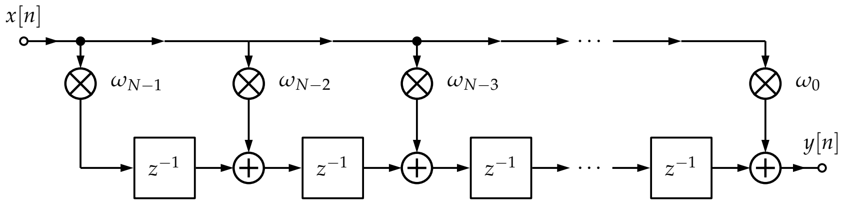


Figure 5.18: Transposed form implementation of a FIR filter.

5.5.2 *Self-heating process FPGA implementation*

For currents modeled as a staircase signal, an accurate thermal response of Equation (3.29) can be evaluated by discretizing the transfer function Equation (3.30) employing the zero-order hold method (Manolakis, 2013, Chapter 1), with the same sample rate as that of the current measurements. The discretization results in a second order discrete $\mathcal{Z}$ transform function in the form of Equation (5.40)

$$H(z) = \frac{\mathcal{Z}\{\theta_s[n]\}}{\mathcal{Z}\{W_c[n]\}} = \frac{\theta_s(z)}{W_c(z)} = \frac{C_{m,1}z + C_{m,2}}{z^2 - C_{d,1}z - C_{d,2}} \tag{5.40}$$

where $W_c[n]$ and $\theta_s[n]$ are the power dissipated in the main cable conductor, not including the sheath and armor losses, and the temperature rise on the cable conductor caused by the self-heating process at discrete-time instant n, and $C_{m,1}$, $C_{m,2}$, $C_{d,1}$ and $C_{d,2}$ are constants that depend on the equivalent RC ladder circuit of the cable model.

Converting Equation (5.40) into difference equations, the architecture shown in Figure 5.19 is obtained. The implementation of the resulting IIR architecture requires 4 parallel multiplications and two sequential additions to be performed for each output sample. This requires for all multipliers and the adder blocks to operate in a single clock cycle

This structure can prevent the sub-block from operating at high clock frequencies, which is desirable for a high throughput pipeline. To this end, a *look-ahead* transformation can be employed to increase the latency of the filter, increasing the total pipeline latency

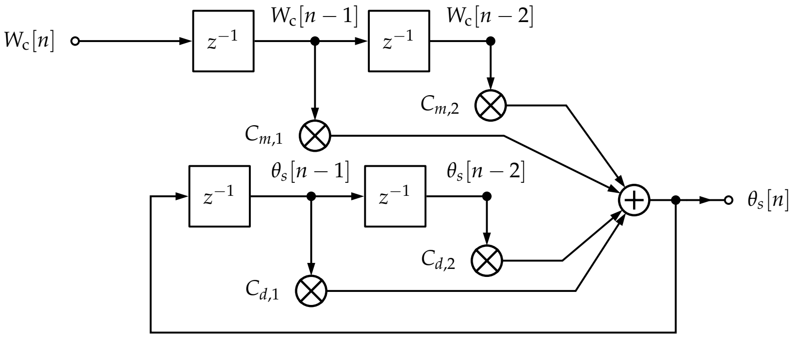


Figure 5.19: Direct implementation of the IIR described by Equation (5.40).

by two extra clocks that permits a fully pipelined architecture (Meyer-Baese, 2014, Chapter 4; Jiang and Willson, 1995; Soderstrand, Sema, and Loomis, 1995)

$$\frac{\theta_s(z)}{W_c(z)} = \frac{C_{W,1}z^3 + C_{W,2}z^2 + C_{W,3}z + C_{W,4}}{z^4 - C_{\theta,3}z - C_{\theta,4}} \tag{5.41}$$

where

$$C_{\theta,3} = C_{d,1}^3 + 2C_{d,1}C_{d,2} \tag{5.42}$$

$$C_{\theta,4} = C_{d,1}^2 C_{d,2} + C_{d,2}^2 \tag{5.43}$$

$$C_{W,1} = C_{m,1} \tag{5.44}$$

$$C_{W,2} = C_{d,1}C_{m,1} + C_{m,2} \tag{5.45}$$

$$C_{W,3} = C_{d,1}^2 C_{m,1} + C_{d,2}C_{m,1} + C_{d,1}C_{m,2} \tag{5.46}$$

$$C_{W,4} = C_{d,1}^2 C_{m,2} + C_{d,2}C_{m,2} \tag{5.47}$$

The transfer function in Equation (5.41) corresponds to the difference equation

$$\begin{aligned}\theta_s[n] = {} & \overbrace{C_{W,1}W_c[n-1] + C_{W,2}W_c[n-2] + C_{W,3}W_c[n-3] + C_{W,4}W_c[n-4]}^{\text{Non-recursive part}} \\ & + \underbrace{C_{\theta_s,3}\theta_s[n-3] + C_{\theta,4}\theta_s[n-4]}_{\text{Recursive part}}\end{aligned} \tag{5.48}$$

Figure 5.20 shows the FPGA implementation of the IIR filter described by Equation (5.48). The numerator part results in a non-recursive filter that can be implemented as a direct FIR design (Meyer-Baese, 2014). A transposed form could have been implemented instead for a lower latency, but it was implemented in a direct form due to the small size and complexity of the filter. The denominator part describes a recursive filter that permits a latency of 3 clocks ($\theta_s[n-3]$ to $\theta_s[n]$) to calculate the output. This extra latency permits calculating both multiplications and the two additions at a higher clock frequency and therefore achieve a higher filter throughput. Due to the latency inherent to the direct FIR filter implementations, registers have been added delaying the enablement of the recursive filter part, in order to synchronize the latencies of both parts as shown in Figure 5.20. Each of the arithmetic blocks in this diagram executes fixed-point operations in a single clock cycle. The pipeline stall, flush and resume mechanism (see Section 5.2) has been implemented by a shifting clock enable signal `CE` that pauses the filter recursion when inputs are invalid, while the sub-blocks with valid inputs and time-independent blocks can still operate. It can be observed that the direct filter implementation operators are not affected by the pipeline stall signals, due to being memoryless and therefore stateless. Their input will only be altered when the enabling signal enables the W_c shift registers, and their output will only affect the IIR section once the enable signal propagates. The output signal `CE`$_{\texttt{out}}$ permits the synchronization with the rest of the pipelined simulator blocks that operate at different latencies as it will be described in Section 5.5.6

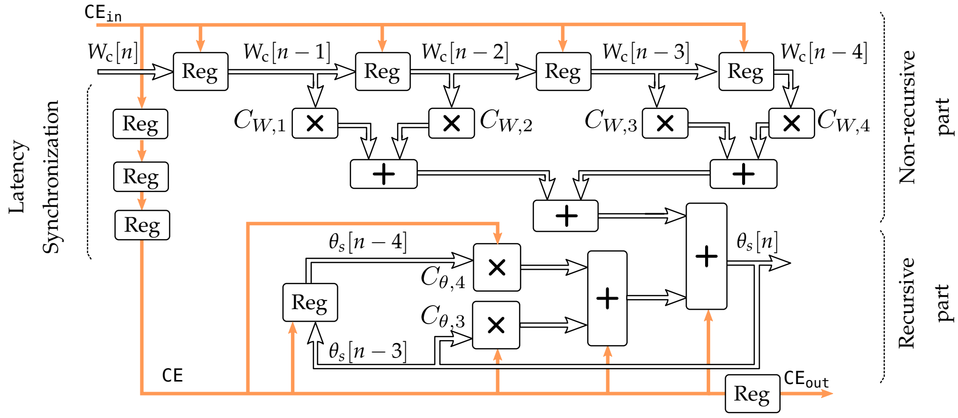


Figure 5.20: Fully pipelined self-heating model implementation.

5.5.3 *Mutual-heating and Soil effect processes*

The mutual-heating process and soil effect are calculated from Equations (3.35) and (3.36). However, those equations describe the thermal response to a constant power applied to the cable. For the estimation of the total thermal response to a varying load the superposition of effects is applied by the convolution as described in Section 3.3.2. In software implementations the exponential integral function is generally approached by a sequential series approximation. However, for hardware logic implementations the complexity and latency of the high resolution divider operators, their sequential operation, and its application to the superposition of the multiple transients covering thousands of hours is impractical.

Both the mutual-heating and the soil effect processes have similar responses that only differ in the constant values affecting them (see the analysis of the responses in Section 5.5.1.2). Therefore, the same approach has been taken for the simulation of both processes, as follows:

Applying Equation (5.27), the discrete-time thermal response of the mutual heating process and the soil effects can be defined by the discrete convolutions

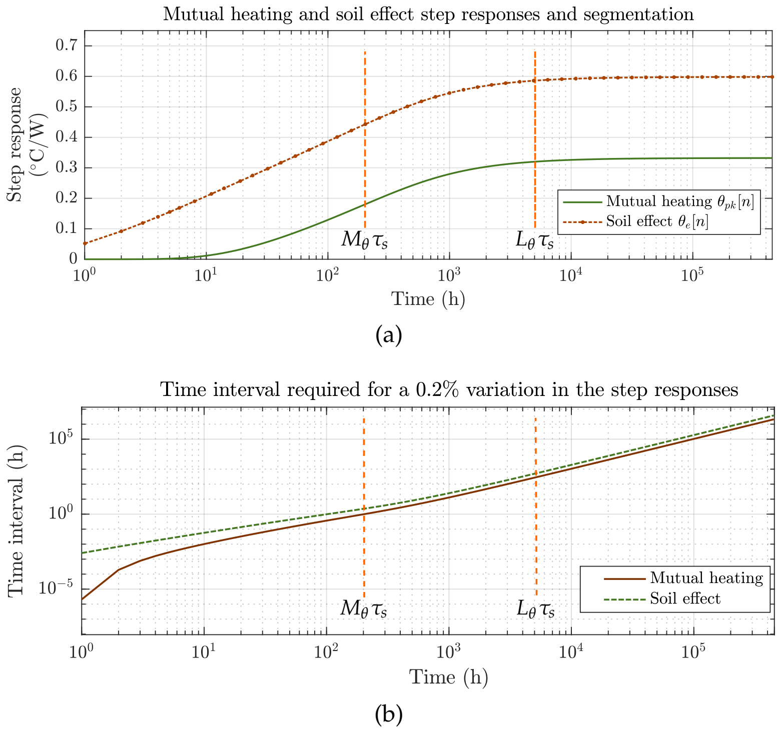
$$\theta_{pk}[n] = \Delta W[n] * \theta_{v,pk}[n] = \sum_{i=0}^{\infty} \Delta W[n-i]\theta_{v,pk}[i] \tag{5.49}$$

$$\theta_{e}[n] = \Delta W[n] * \theta_{v,e}[n] = \sum_{i=0}^{\infty} \Delta W[n-i]\theta_{v,e}[i] \tag{5.50}$$

where $\theta_{pk}[n]$ and $\theta_{e}[n]$ are the discrete-time reponses for the temperature rises at discrete-time instant n due to the mutual heating and soil effects, $\Delta W[n]$ is the dissipated power increment at discrete-time instant n, and $\theta_{v,pk}[n]$ and $\theta_{v,pk}[n]$ are the unit step responses of the mutual heating process and the soil effect. However, due to the long time required for $\theta_{v,pk}[n]$ and $\theta_{v,e}[n]$ to reach steady-state conditions, the convolution would require a large size FIR filter involving thousands of multiply-add logic blocks, which is unfeasible in current FPGA technology.

The implemented architecture takes advantage of the characteristics of the mutual heating and soil effect processes described in Section 5.5.1.2 and the method to segment a filter shown in Section 5.5.1.3.

Figure 5.21a shows the step responses for the Cable A setup in logarithmic time scale, and Figure 5.21b describes the time increment required for a 0.2 % change of the step responses shown in Figure 5.22.



(a)

(b)

Figure 5.21: Mutual heating process and soil effect step responses: (a) Segmentation of the step response and (b) Time interval required for the thermal step responses to increase by 0.2 %.

It can be observed that both the soil and mutual heating processes grow rapidly in the initial transition hours up to the point where $t = M_\theta \tau_s$ defined in Section 5.5.1.2. Within that range, the time interval required for the thermal response to increment by 0.2 % is 1 h, which means that if the input power is sampled every hour, it is necessary to represent every sample individually. However, after those M_θ hours, the growing rate decreases geometrically, and after L_θ hours it reaches to a near steady-state condition.

If the response curve between $t = M_\theta \tau_s$ and $t = L_\theta \tau_s$ is divided into sub-segments of the length given by Figure 5.21b, any error due to approximating the curve to its average value will not be larger than 0.1 %, or 0.2 % if the value is rounded to either edge.

It is important to notice that the values of M_θ and L_θ differ for the mutual heating and the soil effect processes. The former can be calculated according to Equation (5.36) and Equation (5.37), and the latter are obtained numerically. However, due to the convenience to merge both filters as it will be described in this section, in practice both can be replaced by the maximum sample index of the two processes.

As shown by Equation (5.38), if the step response is divided into segments, the last segment must reproduce an infinite impulse response, which can be implemented by an IIR filter. A possible approach to obtain a recursive filter structure would be considering that the last filter stage had reached the steady-state condition of the thermal process, resulting constant filter coefficients of value $\theta_v(\infty)$, and its convolution would become

$$\delta[n - P_N] * \Delta W[n] * \theta_v(\infty) = \theta_v(\infty) \sum_{i=P_N}^{n} \Delta W[n - i] \tag{5.51}$$

which describes an infinite tap filter with a single constant coefficient.

However, it was found that assuming that $\theta_{v,pk}[n]$ or $\theta_{v,e}[n]$ reached the steady-state from time instant $L_\theta \tau_s$ can cause unacceptable estimation errors for the desired accuracy, even with $L_\theta \tau_s$ in the order of thousands of hours. This effect can be observed in Figure 5.10 that shows the difference of the thermal response with respect to the steady-state conditions.

When trying to represent this process by a Laplace transform for its discretization, it can be observed that the thermal impulse responses in this region follow the form $\mathrm{e}^{\frac{\kappa_1}{t}} - \mathrm{e}^{\frac{\kappa_2}{t}}$ where κ_1 and κ_2 are constants (see Equations (5.13) and (5.15)) which have no Laplace transform. Even an approximate function for its first term in the Taylor series for larger values of t as $\frac{\kappa_1 - \kappa_2}{t}$ results in the non-linear z-transform $(\kappa_1 - \kappa_2) \ln\left(\frac{z}{z-1}\right)$ defined only for samples $n > 0$.

However, it was found that the temperature increment in the step responses $\theta_v[n]$ (either $\theta_{v,pk}[n]$ or $\theta_{v,e}[n]$) after certain L_θ time samples can be approximated by the step response of a first order process delayed L_θ samples in time as

$$\theta_v(\infty) - \theta_v(t) \approx (\theta_v(\infty) - \theta_v(L_\theta \tau_s))(1 - \mathrm{e}^{-\frac{t - L_\theta \tau_s}{\tau_c}}) \tag{5.52}$$

where τ_c is the time constant of the first order process and τ_s is the discrete sampling time interval. Or in discrete form:

$$\theta_v[\infty] - \theta_v[n] \approx (\theta_v[\infty] - \theta_v[L_\theta])(1 - \mathrm{e}^{-\frac{n\tau_s - L_\theta \tau_s}{\tau_c}}) \tag{5.53}$$

The corresponding z-transform function is (Oppenheim et al., 1996, Chapter 10):

$$\mathcal{Z}\{\theta_v[\infty] - \theta_v[n]\} = z^{-L_\theta} \left(\theta_v[\infty] - \theta_v[L_\theta]\right) \frac{z(1 - \mathrm{e}^{\tau_s/\tau_c})}{\tau_c^{-1}(z-1)(z - \mathrm{e}^{-\tau_s/\tau_c})} \tag{5.54}$$

Accordingly, after L_θ samples, $\theta_{v,pk}[n]$ and $\theta_{v,e}[n]$ can be approximated by the processes $\theta_{v,pk,L}[n]$ and $\theta_{v,e,L}[n]$, that are described by a first order response plus a step function that reaches to the steady-state condition. Therefore, for $n \geq L_\theta$, Equation (3.36) can be discretized as

$$\theta_{v,pk}[n] \approx \hat{\theta}_{v,pk,L}[n - L_\theta], \quad n \geq L_\theta \tag{5.55}$$

$$\mathcal{Z}\{\hat{\theta}_{v,pk,L}[n]\} = \frac{\theta_{v,pk}[L_\theta] z}{z-1} + (\theta_{v,pk}[\infty] - \theta_{v,pk}[L_\theta]) \frac{z}{z-1} \cdot \frac{1 - e^{-\tau_s/\tau_{pk}}}{z - e^{-\tau_s/\tau_{pk}}} \tag{5.56}$$

where τ_s is the sampling interval and τ_{pk} is the time constant of the first order response for the last segment of the step response of the mutual heating process.

Similarly, for the soil effect response at $n \geq L_\theta$ (where the length has been calculated for the soil effect response), Equation (3.35) can be discretized as

$$\theta_{v,e}[n] \approx \hat{\theta}_{v,e,L}[n - L_\theta], \quad n \geq L_\theta \tag{5.57}$$

$$\mathcal{Z}\{\hat{\theta}_{v,e,L}[n]\} = \frac{\theta_{v,e}[L_\theta] z}{z-1} + (\theta_{v,pk}[\infty] - \theta_{v,pk}[L_\theta]) \frac{z}{z-1} \cdot \frac{1 - e^{-\tau_s/\tau_e}}{z - e^{-\tau_s/\tau_e}} \tag{5.58}$$

where τ_e is the time constant of the first order response for the last segment of the step response of the soil effect.

For the samples between M_θ and L_θ, the discrete convolution sum of the thermal process can be split into multiple sub-segments as the sum of delayed responses, according to Equation (5.38). Based on this method, the convolution that represents the thermal responses of the mutual-heating and soil effect can be divided into 3 segments as shown in Figure 5.22:

i) An initial FIR filter covering the first $M_\theta \tau_s$=200-300h of transition, where each of the weights correspond to samples from Equations (3.35) and (3.36);

ii) a segmented FIR filter set covering $L_\theta - M_\theta$ samples depending on the cable structure; and

iii) an IIR filter for the response after that time instant, described by Equation (5.54).

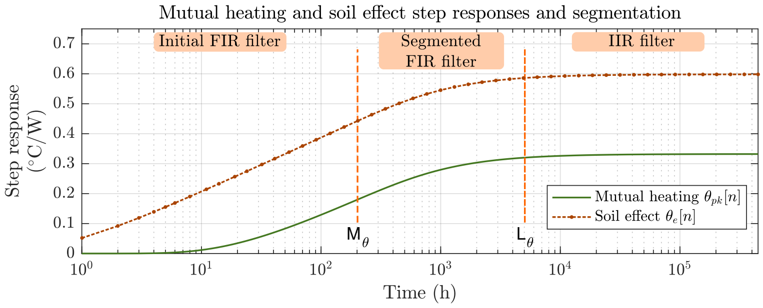


Figure 5.22: Segmentation of the mutual and soil step responses.

5.5.3.1 *First segment implementation of the Mutual-heating and Soil effect processes*

For the first segment FPGA implementation, the synthesis of a direct filter architecture was found not be generally feasible within the given latency constraints, due to its large filter size that makes routing complex. The transposed filter architecture described in Section 5.5.1.4 is suggested instead. In order to pipeline the structure, an enablement signal `CE` is carried through the filter as shown in Figure 5.23. The pipeline latency of the filter is of 2 clocks-cycles, and has a throughput of 1 sample per clock cycle. The input to the filter is the output of the other filter segment implementations, because, as it will be described later in this section (see Equation (5.69)), the full process response is obtained by delaying each filter segment and adding the responses. This is equivalent to chaining the output of the next segments or stages (s_1 in this case) as shown in Figure 5.26. In this implementation the weights, denoted $\theta_v[i]$ are equal to the sample values from the mutual heating and soil effect step responses, either $\theta_v[i] = \theta_{v,pk}[i] = \theta_{v,pk}(\tau_s i)$ for the mutual-heating process, or $\theta_v[i] = \theta_{v,e}[i] = \theta_{v,e}(\tau_s i)$ for the soil effect process.

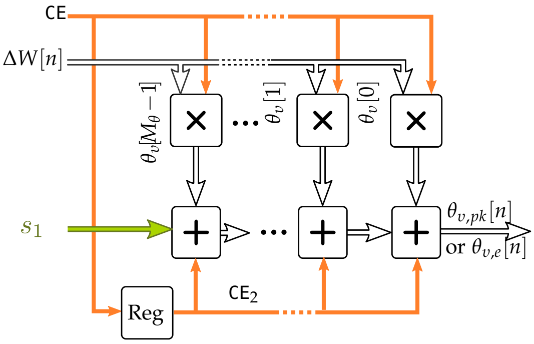


Figure 5.23: First segment implementation as a transposed FIR filter.

5.5.3.2 *IIR segment implementation of the Mutual-heating and Soil effect processes*

The last stage IIR filter is achieved by an implementation of the difference equations of the discrete transfer function in Equation (5.58). Merging the equation into a 3^{rd} order transfer function is not suggested, as it can lead to rounding errors and increased latency.

Therefore, it has been implemented by obtaining the difference equations of the transfer function in three parts and converting each part into the blocks marked in Figure 5.24, where for the mutual-heating process

$$G_1 = \theta_{v,pk}[\infty] - \theta_{v,pk}[L_\theta] \tag{5.59}$$

$$G_2 = \theta_{v,pk}[L_\theta] \tag{5.60}$$

and for the soil effect process

$$G_1 = \theta_{v,e}[\infty] - \theta_{v,e}[L_\theta] \tag{5.61}$$

$$G_2 = \theta_{v,e}[L_\theta] \tag{5.62}$$

The total pipeline latency of the filter, shown in Figure 5.24 is $\tau_{\text{pip}} = 3$ clocks. The different clock enabling signals `CE`, `CE`$_2$ and `CE`$_3$ permit controlling the pipeline enablement at the different stages. The architecture uses three multiply-accumulate structures, which operate in a single clock cycle (a block latency of a single clock). These blocks have been identified by grouping filled areas in the figure.

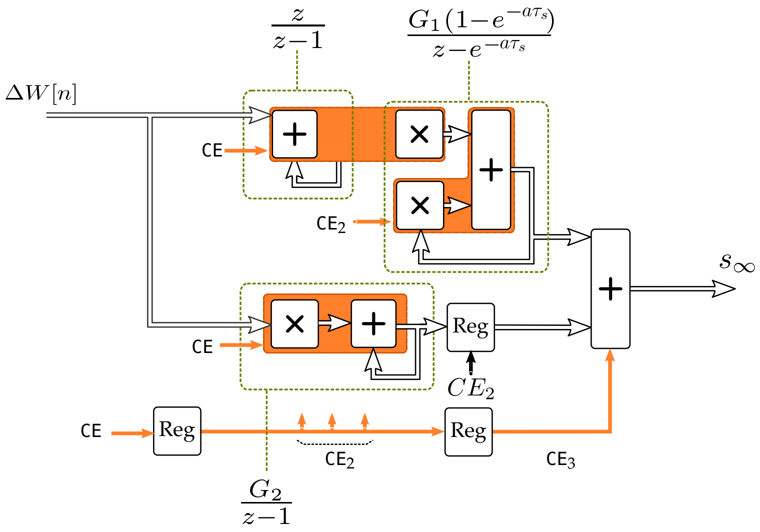


Figure 5.24: IIR filter implementation for the last segment.

5.5.3.3 *Segmented FIR implementation of the Mutual-heating and Soil effect processes*

According to the segmentation shown in Figure 5.22, the convolutions in Equation (5.49) and Equation (5.50) are approximated as

$$\theta_{pk}[n] \approx \sum_{i=0}^{L_\theta-1} \Delta W[n-i]\theta_{v,pk}[i] + \sum_{i=L_\theta}^{\infty} \Delta W[n-i]\hat{\theta}_{v,pk,L}[i-L_\theta] \tag{5.63}$$

$$\theta_{e}[n] \approx \sum_{i=0}^{L_\theta-1} \Delta W[n-i]\theta_{v,e}[i] + \sum_{i=L_\theta}^{\infty} \Delta W[n-i]\hat{\theta}_{v,e,L}[i-L_\theta] \tag{5.64}$$

where the second term in both is replaced by the IIR filter in Figure 5.24 and the first term represents a FIR filter of size L_θ.

However, the size L_θ required for a given accuracy of Equation (5.58) can be too large to fit in an FPGA. The necessary number of samples L_θ is dependent on the time constants defined by the cable dimensions, soil thermal characteristics, and the cable setup. Tests have shown that, for cable setups with large time constants and $L_\theta = 5000$ samples at

$\tau_s = 1\,\mathrm{h}$, a maximum deviation of 0.4 % was achieved by the first order approximation Equation (5.58) for the mutual heating process, and a maximum error of 0.2 % for the soil effect.

The implementation of this filter segment can be achieved by splitting the filter into a set of sub-filters, and taking advantage of the decreasing derivatives of $\theta_{pk}(t)$ and $\theta_e(t)$. From Equation (5.38)

$$\begin{aligned}&\sum_{i=M_\theta}^{L_\theta-1} \Delta W[n-i]\theta_v[i]\\ &\qquad = \sum_{i=M_\theta}^{P_1-1} \Delta W[n-i]\theta_v[i] + \sum_{i=P_1}^{P_2-1} \Delta W[n-i]\theta_v[i] + \cdots + \sum_{i=P_{N-1}}^{L_\theta-1} \Delta W[n-i]\theta_v[i]\end{aligned} \tag{5.65}$$

where $P_1, P_2 \ldots, P_N$ are the sample time indexes where the filter is split.

When the filter weights for each segment j are approximated with a constant value $\hat{\theta}_v^{(j)}$ for all time indexes in the segment,

$$\theta_v[i] \approx \hat{\theta}_v^{(j)} \forall i \in P_{j-1} \ldots P_j - 1 \tag{5.66}$$

the filter can be rewritten as

$$\begin{aligned}&\sum_{i=M_\theta}^{L_\theta-1} \Delta W[n-i]\theta_v[i]\\ &\qquad \approx \hat{\theta}_v^{(1)} \sum_{i=M_\theta}^{P_1-1} \Delta W[n-i] + \hat{\theta}_v^{(2)} \sum_{i=P_1}^{P_2-1} \Delta W[n-i] + \cdots + \hat{\theta}_v^{(N-1)} \sum_{i=P_N}^{L_\theta-1} \Delta W[n-i]\end{aligned} \tag{5.67}$$

Due to the geometrically decreasing derivative function of the thermal response, the filter segments are growing size and this size is calculated numerically for the required accuracy. Each of the segments described by Equation (5.67) describes an accumulate and multiply process that is applied to a sliding window covering q_j samples of the input $\Delta W[n]$. The sliding window can be implemented by means a shift register as shown in Figure 5.25. The shift register carries inputs multiplied by the coefficient $\hat{\theta}_v^{(j)}$, then a block introduces (adds) the samples when they enter the given latency of the window, and removes (subtracts) them when they exit the window (and enter the next segment). An additional shift-register is required for the addition of the rest of the segments, which have to be added at different latencies, according to Equation (5.67).

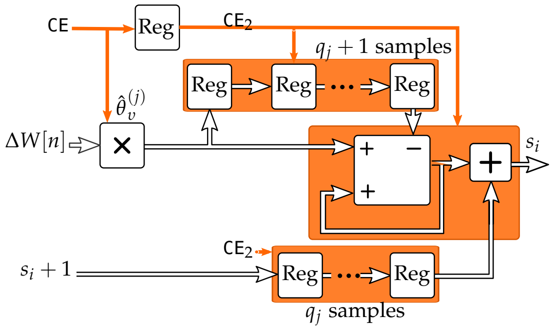


Figure 5.25: Implementation of each FIR sub-filter segment.

Each stage is then converted as follows, into simplified sub-filter architectures as shown in Figure 5.26.

The segmented filter set contains a set of q FIR filters in total, that have large $\{m_1, m_2, \ldots, m_j, \ldots m_q\}$ filter lengths that grow geometrically with j. Each j^{th} filter has a constant coefficient value $\hat{\theta}_v^{(j)}$ for all its weights ($\hat{\theta}_{v,pk}^{(j)}$ for the mutual heating process and $\hat{\theta}_{v,e}^{(j)}$ for the soil effect process), and follows the structure of Figure 5.25.

The effect of the varying filter segment sizes is equivalent to increasing geometrically the sampling time of $\theta_{v,pk}[n]$ and $\theta_{v,e}[n]$, while the sampling rate of the input power transitions $\Delta W[n]$ is maintained (because all transitions have an effect on the future thermal response). In between $\theta_{v,pk}[n]$ or $\theta_{v,e}[n]$ samples, filter values are kept constant.

5.5.3.4 *Full filter implementation of the Mutual-heating and Soil effect processes*

Merging the segmented FIR filter implementation in Section 5.5.3.3 and the IIR filter $\mathcal{Z}\{\hat{\theta}_{v,pk,L}[n]\}$ described in Equation (5.58) as shown in Section 5.5.3.2, the mutual-heating process can be implemented as follows:

$$\begin{aligned}\theta_{pk}[n] \approx &\underbrace{\sum_{i=0}^{M-1} \Delta W[n-i]\theta_{v,pk}[i]}_{\text{FIR}}\\ &+ \underbrace{\hat{\theta}_{v,pk}^{(1)} \sum_{i=M}^{M+m_1-1} \Delta W[n-i] + \ldots + \hat{\theta}_{v,pk}^{(q)} \sum_{\substack{i=M+m_1\\ +m_2+\cdots+m_{(q-1)}}}^{\substack{M+m_1+m_2\\ +\cdots+m_q-1}} \Delta W[n-i]}_{\text{set of } q \text{ FIR subfilters}}\\ &+ \underbrace{\sum_{i=L_\theta}^{\infty} \Delta W[n-i]\hat{g}_{pk,L}[i-L]}_{\text{IIR}}\end{aligned} \tag{5.68}$$

and the soil effect process as

$$\begin{aligned}\theta_{e}[n] \approx &\underbrace{\sum_{i=0}^{M-1} \Delta W[n-i]\theta_{v,e}[i]}_{\text{FIR}}\\ &+ \underbrace{\hat{\theta}_{v,e}^{(1)} \sum_{i=M}^{M+m_1-1} \Delta W[n-i] + \ldots + \hat{\theta}_{v,e}^{(q)} \sum_{\substack{i=M+m_1\\ +m_2+\cdots+m_{(q-1)}}}^{\substack{M+m_1+m_2\\ +\cdots+m_q-1}} \Delta W[n-i]}_{\text{set of } q \text{ FIR subfilters}}\\ &+ \underbrace{\sum_{i=L_\theta}^{\infty} \Delta W[n-i]\hat{g}_{e,L}[i-L]}_{\text{IIR}}\end{aligned} \tag{5.69}$$

This method permits reducing the L_θ multiply-accumulate blocks required for the filter in Equation (5.64), into an architecture of q sub-filters with a single $\hat{\theta}_{v,pk}^{(j)}$ multiplier and an accumulator for each stage j, as shown in Figure 5.25.

In practice, if the power of the cable of interest and the side cables carry the same load or are proportional to each other, both filter-sets (the mutual-heating and soil effect filters) can be merged into a single filter-set, by using equally sized segments for both processes and adding their weights. This can save implementation resources, reduce routing issues and produce a much more efficient architecture. However, during this thesis both filters have been implemented separately, in order to be able to analyze and validate each process.

The complete model implementation is shown in Figure 5.26. The input to the model is the total dissipated power in the heat source. That is, the total dissipated power within the cable under analysis $W_{\text{tot},s}$ for the soil effect process, and the total dissipated power in each adjacent cable $W_{\text{tot,side}}$ for the mutual heating process. While the filter sizes and coefficients for each filter stages vary, the structure is the same for both models. The power transitions $\Delta W_{\text{tot},s}$ and $\Delta W_{\text{tot,side}}$ are calculated by a latch register and a subtractor block, which are fed directly to each of the filter stages.

The initial filter and each stage filter outputs are connected to the next stage filter inputs, and the IIR filter from Equation (5.58) (Figure 5.24) is connected as the last stage. The total input to output latency is 3 clocks with a throughput of 1 sample per clock for consecutive outputs. Delay registers labeled *sync* permit the synchronization of the outputs due to the differences in latency between the segments.

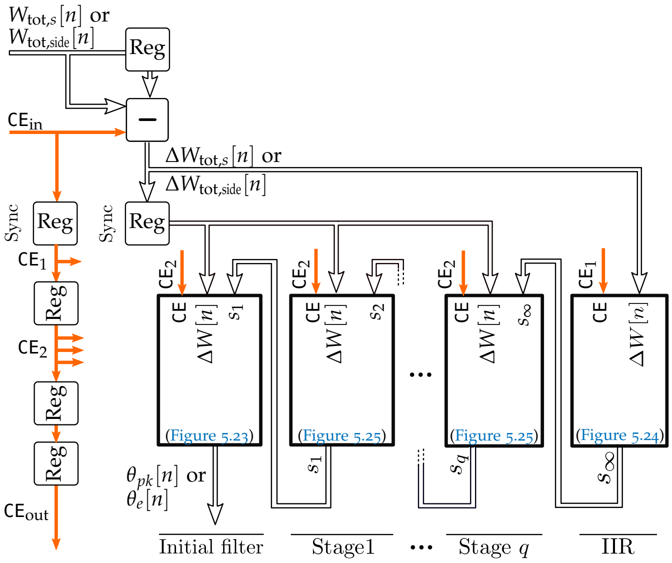


Figure 5.26: Complete pipelined hardware logic implementation of the mutual heating and soil effect models.

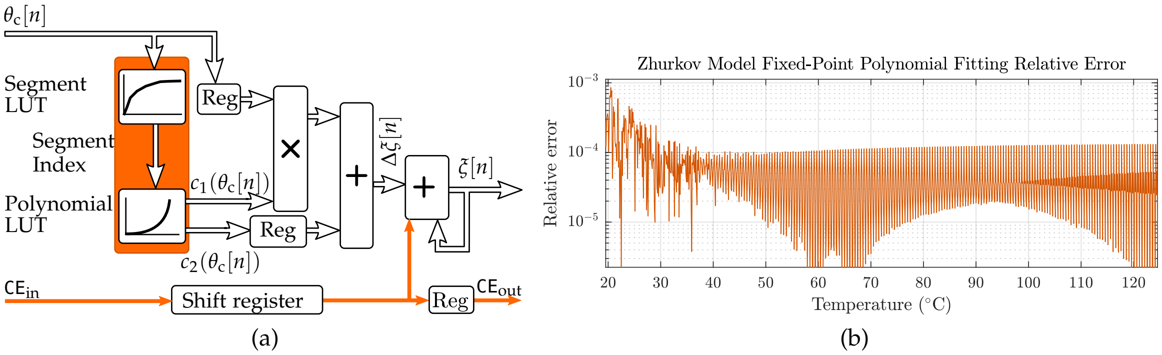
5.5.4 *Thermal degradation model*

The degradation function described by Equation (3.58) grows exponentially with the temperature. A reduced-latency implementation for this non-linear function was achieved by segmenting the curve and approximating each segment by a linear regression. The implemented architecture is shown in Figure 5.27a. Larger polynomial approximations are also possible, but a first order solution was found to suffice for the required accuracy. The degradation increment $\Delta\xi$ for each time interval of $\Delta t = \tau_s$ hours, is then estimated following the Palmgren and Miner rule in Equation (3.59) as

$$\Delta\xi(\theta, n) \approx \frac{\tau_s}{\tau\left(\theta(n)\right)} \approx c_1\left(\theta_c[n]\right) \cdot \theta_c[n] + c_2\left(\theta_c[n]\right) \tag{5.70}$$

where c_1 and c_2 are the coefficients of the polynomial for each segment of the curve. The polynomial is calculated by the stateless pipelined multiply-add blocks in Figure 5.27a.

The maximum temperature of the insulator or the temperature of the main conductor $\theta_c[n]$ is input to a first stage LUT that indexes the segment of the curve. A non-linear segmentation permits increasing the resolution in high derivative areas of the function for better accuracy, and permits reducing the FPGA LUT resources required to implement polynomial indexing and to record their coefficients.



(a) (b)

Figure 5.27: Mutual and soil thermal step responses: (a) pipelined Zhurkov model architecture implementation, and (b) relative error of the polynomial fitting of the Zhurkov Model.

A second LUT obtains the polynomial coefficients c_1 and c_2 corresponding to the segment index, that approximate Equation (3.58).

The accumulator block in the last stage in Figure 5.27a accumulates the $\Delta\xi(\theta_c[n])$ increments following Equation (3.61). Fixed point operators of 44 bit fraction resolution were necessary for the Cable A setup (Figure A.1) degradation model. All blocks except the accumulator are memoryless (stateless). Therefore, the pipeline clock enablement signals only affect the accumulator block. The enablement signal can be carried through to the other blocks, but that would increase the complexity of the routing unnecessarily. The test results on this cable setup showed that an accuracy of 0.1 % of the Zhurkov model is achieved by this fixed-point pipelined method, in comparison to the floating-point results, as shown in Figure 5.27b.

For simplicity, the electrical field was assumed to be constant for this implementation, but a two-dimensional LUT and interpolation can be similarly used for varying electrical field cases.

It is important at this point to notice that the temperature estimation deviations will impact the estimations of the degradation increments. The degradation curves are highly non-linear and very sensitive to temperature values and slight value changes can cause large degradation value changes at each estimation interval. Because of this, the mutual-heating and the soil effect thermal models were designed with a 0.1 % to 0.2 % accuracy. Figure 5.28 shows the error caused on the degradation increment by deviation percentage in the temperature estimation.

It can be observed that due to the non-linearity of the degradation function, the effect of the temperature estimation errors is much larger when the cable is at higher temperatures than at lower temperatures. Even direct measurements of the cable temperature by common means such as fiber optics, generally with a 1% accuracy in the measurements, cannot offer such accuracies on the temperature values (Yilmaz and Karlik, 2006). Therefore, the models employed through this thesis consider higher resolutions than actual measurements. However, the impact of the thermal model deviations needs to be considered when interpreting the results.

5.5.5 *Currents and Dissipated Power*

In order to validate the model, real load patterns need to be injected into the model, and the total power dissipated in each cable must be evaluated, in order to calculate the mutual-

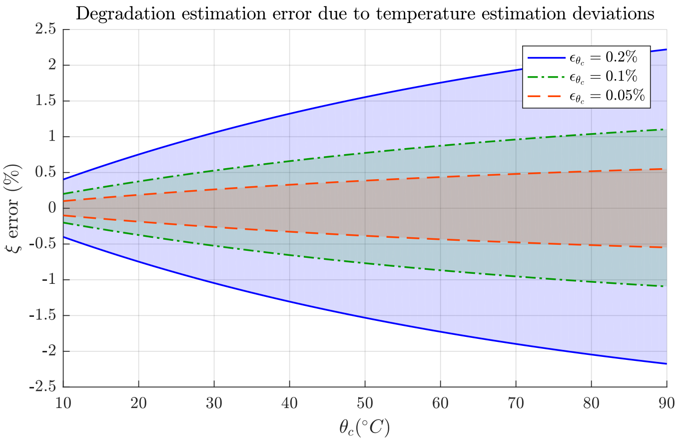


Figure 5.28: Effect of $\pm\epsilon_{\theta_c}\%$ deviations in the temperature estimation on the estimation of the degradation of the cable insulator.

heating and soil effect thermal responses. Figure 5.29 shows the implementation for the models of currents and dissipated power.

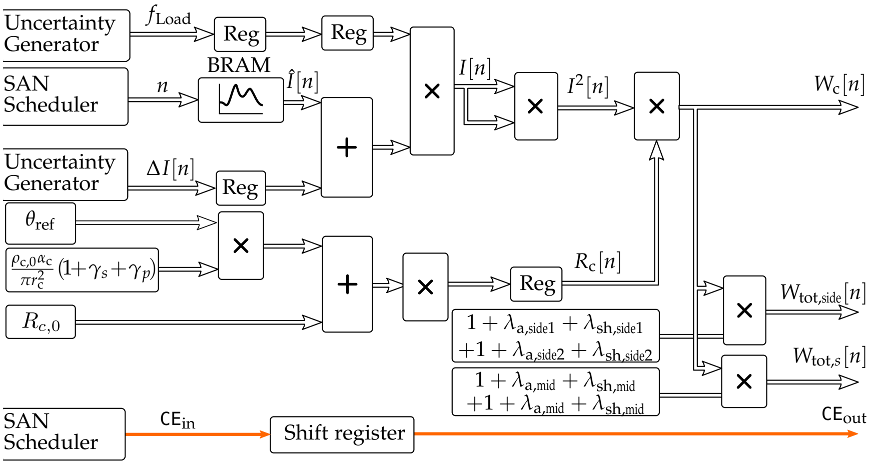


Figure 5.29: Pipelined input power sequence generation logic.

In order to model a full year load, a Block RAM (BRAM) is used to store a full year hourly load values, which were obtained from (Red Eléctrica de España, 2019) and scaled to the nominal currents of the Cable A in Appendix A. These load patterns were assumed to be constant over the years for this model. However, if a trend model is known for the specific grid, a multiplier can be employed for its scaling (Bicen, 2017).

In practice, the load values stored in BRAM can represent both past measurements for Condition Monitoring (CM) or future predicted load values for performing a prognosis. However, the future loads will also depend on the switching and component state models from the SAN grid model that controls the load factor f_{Load} applied the load patterns, by a multiplier input in Figure 5.29.

In either case, both the measurements and future predictions are affected by measurement noise and deviations in the predictions, and both must be considered when predicting the cable Remaining Useful Life (RUL).

These uncertainties are generated by an external *uncertainty generator*. This block models the prediction deviations and/or measurement noise based on a Pseudo-Random Number Generation (PRNG) that calculates $\Delta I[n]$ deviations in the load applied to the cable. The block also injects a varying f_{Load} which is random and depending on the grid condition given by the SAN model.

As described in Section 3.3.2, the varying resistivity of the conductors with temperature can be taken into account by calculating the results according to Equation (3.40). Accordingly, the calculations of the instantaneous dissipated power is performed for a reference temperature θ_{ref} and the conductor radius and compensated after the thermal calculations have been performed. The reference conductor resistivity can be directly introduced as parameter in the model. However, for validation and flexibility purposes, the resistance is calculated from a reference Alternating Current (AC) resistance $R_{\mathrm{c},0}$ and reference resistivity values $\rho_{\mathrm{c},0}$ at 0 °C, and a factor that includes based a reference conductor resistivity $\rho_{\mathrm{c},\theta_{\mathrm{ref}}}$, the temperature coefficient α_{c}, and skin and proximity effect factors γ_s and γ_p (Rerak and Ocłoń, 2017).

The block generates separate output values for the power dissipated in the main conductor W_{c}, and the total power dissipated in the middle cable $W_{\mathrm{tot},s}$ and in adjacent cables $W_{\mathrm{tot,side}}$. Cable load values are read from a BRAM memory in the programmable logic part of the FPGA, as shown in Figure 5.29. In the implementation used through this thesis hourly full-year load values are recorded. The BRAM can be accessed and preloaded by the CPU for each simulation, in order to include past current measurements and updated predicted future load patterns.

The sheath and armor loss factors for the middle cable $\lambda_{\mathrm{a,mid}}$ and $\lambda_{\mathrm{a,mid}}$ and for each k^{th} side cable $\lambda_{\mathrm{a,side}k}$ and $\lambda_{\mathrm{sh,side}k}$, are included as multiplying parameters for adapting the model to different cables or varying parameters. The loss factors can be included within the mutual and soil heating model filters described in Section 5.5.3 for a reduced size implementation, at the cost of a reduced flexibility.

The discrete-time index n and the clock enable signal $\mathtt{CE}_{\mathrm{in}}$ are driven by the SAN scheduler, which stalls and flushes the pipeline every time the grid state changes. However, in the dissipated power model all the blocks in the pipeline are memoryless and therefore they are not affected by the pipeline pause and flushing.

5.5.6 *Full thermal simulation architecture and interfacing to the SAN MC Simulator*

The total cable temperature is calculated by the addition of all the thermal heating processes, as shown in Figure 5.30. The resulting temperature $\hat{\theta}_{\mathrm{c}}$ includes already the attainment factor in the mutual and soil heating processes, calculated for resistivity values at temperature θ_{ref}.

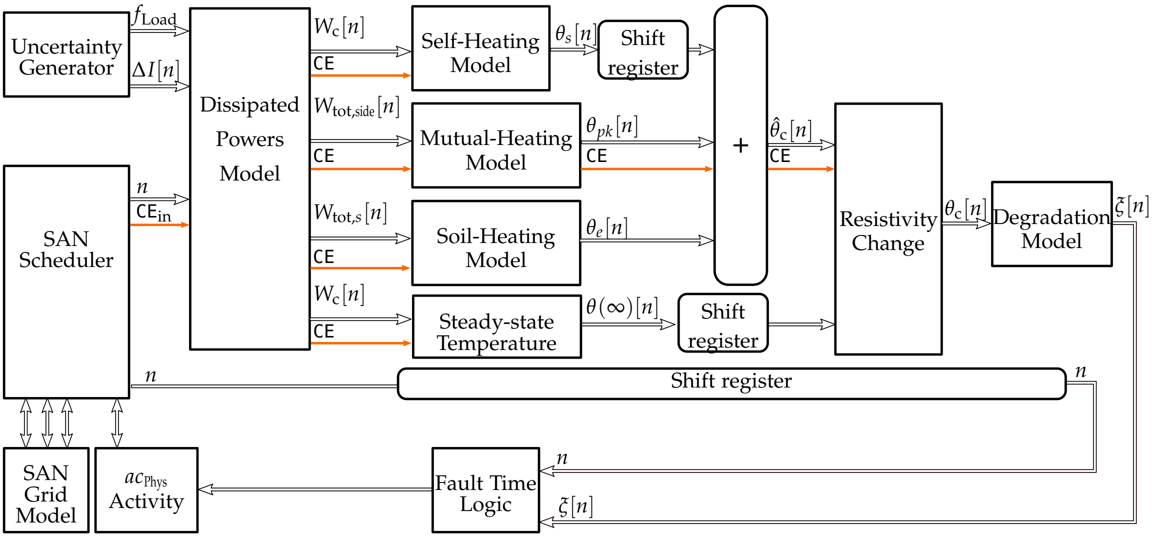


Figure 5.30: Full FPGA Grid and Cable Degradation simulator architecture.

However, the resulting temperature needs to be compensated for the change of resistivity with the cable temperature according to Equation (3.40). Accordingly, it is necessary to calculate the maximum temperature rise in steady state conditions $\theta_M(\infty)$ and $\theta_M(t)$ that is caused by the constant heat rate at steady-state.

When the load of all the cables in the set are equal or proportional to each other, the steady-state temperature rise in the cable $\theta_M(\infty)$ is proportional to power dissipated power in the main conductor in steady-state $W_{\mathrm{c},R_{\theta(\infty)}}$, by a constant C. This constant depends on the cable-set structure, and it is obtained from the steady-state equations for self-heating, mutual-heating and soil effect processes given by Equations (3.30), (3.44) and (3.45) and the loss factors that relate the power dissipated in the main conductor with the total dissipated power. For equally loaded cables:

$$C = \mathcal{T}_{\mathrm{A}} + \mathcal{T}_{\mathrm{B}} + (1 + \lambda_{\mathrm{sh,mid}} + \lambda_{\mathrm{a,mid}}) \frac{\varrho_{\mathrm{s}}}{2\pi} \ln \frac{4L_d}{D_e} + \sum_{k \in \mathcal{S}} (1 + \lambda_{\mathrm{a,side}k} + \lambda_{\mathrm{sh,side}k}) \frac{\varrho_{\mathrm{s}}}{2\pi} \ln \frac{d'_{pk}}{d_{pk}} \tag{5.71}$$

Therefore, in steady-state:

$$\theta_{\mathrm{c}}(\infty) - \theta_{\mathrm{ref}} = W_{\mathrm{c},R_{\theta(\infty)}} C \tag{5.72}$$

and due to the increase of the thermal resistivity with temperature, it can be related to the power calculated for the same current load at the reference temperature θ_{ref}

$$\theta_{\mathrm{c}}(\infty) - \theta_{\mathrm{ref}} = W_{\mathrm{c},R_{\theta_{\mathrm{ref}}}} (1 + \alpha_{\mathrm{c}}(\theta_{\mathrm{c}}(\infty) - \theta_{\mathrm{ref}}))C \tag{5.73}$$

resulting

$$\theta_{\mathrm{c}}(\infty) - \theta_{\mathrm{ref}} = \frac{C W_{\mathrm{c},R_{\theta_{\mathrm{ref}}}}}{1 - W_{\mathrm{c},R_{\theta_{\mathrm{ref}}}} \alpha_{\mathrm{c}} C} \tag{5.74}$$

where $W_{\mathrm{c},R_{\theta_{\mathrm{ref}}}}$ is the steady state power calculated at the reference temperature θ_{ref}.

The *Steady-state Temperature* block in Figure 5.30 implements Equation (5.74) using a pipelined divider Intellectual Property (IP) block by Xilinx, with a total of 22 clock pipeline latency, and a throughput of 1 sample per clock.

Similarly, the resistivity change compensation block applies equation Equation (3.40), requiring a 26 clock latency pipeline.

A simple fault logic block compares the resulting degradation level $\xi[n]$ with the acceptable degradation level, which was set to be < 1.0, or $< 100\,\%$ in this application, and produces a failure time value to the physical model activity ac_{Phys} block.

5.5.7 *Performance of the architecture*

In order to validate the thermal modeling architecture and its results, the results were compared against existing literature and standards.

Figure 5.31 shows the thermal response for a single trefoil cable setup shown in Cable-Setup B in Appendix A.2, which was analyzed in (Diaz-Aguiló and León, 2015). The results show that the model accurately follows the results obtained by the authors in (Diaz-Aguiló and León, 2015).

However, no long-term numerical thermal response data was found for comparison in the existing literature. Therefore, the architecture was tested on the Cable A setup shown in Figure A.1, and a full convolutional method was compared against the methods described in this chapter. For this implementation, $M_\theta = 200$ samples were required for the initial

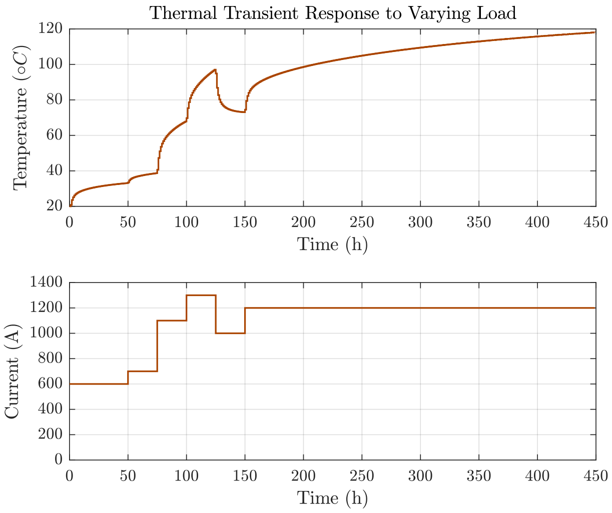


Figure 5.31: Thermal response of Cable-Setup B setup from (Diaz-Aguiló and León, 2015).

filter segments of both the mutual-heating processes and the soil effect models, q=218 second stage filters were required for a 0.2% accuracy of the mutual heating, and q=114 for the soil effect. If better resolutions are required, a linearly interpolated coefficient set can also be used with extra multiply-accumulate blocks, with the associated increase of the resource requirements.

The architecture was synthesized for a XCZU9EG-2 Xilinx FPGA, and the resulting resource usage and timing analysis results are shown in Table 5.1. Notice that the resource utilization can be further reduced by merging both mutual heating and soil effect filters into a single filter set. In this implementation both filters were synthesized separately, for testing, analysis, and validation purposes.

Table 5.1: FPGA resource usage by the electro-thermal degradation model and timing analysis

FPGA Device	XCZU9EG-2	
PL Clock	187 MHz	
Thermal Margin@25.0 °C	68.4 °C	
Resource	**Count**	**(%)**
LUT	137231	50.07
Registers	94529	17.24
DSP	1612	63.97
BRAM Tile	26.5	2.91
Timing analysis results	**Value**	
Worst Negative Slack (WNS)	0.090 ns	
Worst Hold Slack (WHS)	0.010 ns	
Worst Pulse Width Slack (WPWS)	1.164 ns	

The architecture permitted an implementation with up to 187 MHz clock frequency, producing a throughput of 1 sample every 5.35 ns, which permits a 50 year simulation with 1 hour samples in just 2.34 ms.

Figure 5.32 shows a result comparison between a floating-point software simulation and the fixed-point FPGA implementation of cable Cable A setup.

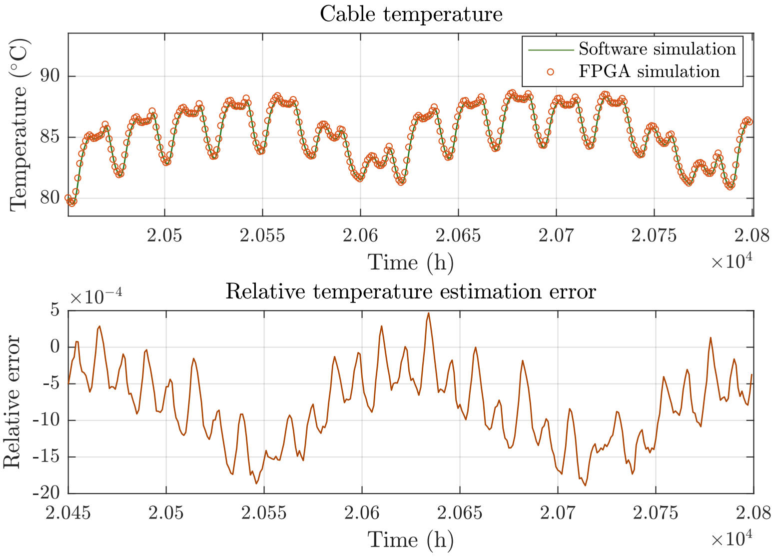


Figure 5.32: FPGA temperature estimation results compared to software simulation.

The maximum error between the models was found to be of 0.5 %, which is caused by the cumulative errors of both the mutual heating process and the soil effect. This error can be reduced if both processes are merged into a single block.

6

DIAGNOSTICS AND PROGNOSIS OF THE CABLE BASED ON FPGA

As discussed in Chapter 1, the deviations in the monitoring process, the errors in the physical models or reduction of these models for their simplification can impact the prognostics results for a cable. This chapter analyzes the effect that those uncertainties cause on the cable lifetime predictions, employing the techniques developed in Chapter 4 and Chapter 5, and the applicability of the methods is evaluated.

It is important to differentiate two separate phases in the analysis of the cable Remaining Useful Life (RUL) and define the different uncertainties that are expected to impact on the cable degradation process:

1. Condition Monitoring (CM)+Diagnostics phase: it covers the period from the time the cable was installed or repaired as-good-as-new, to the instant t_0 when the prognostics analysis takes place;
2. Prognostics phase: it covers the period from t_0 until the end time of the prognosis.

Through this thesis, physics-based models have been adopted for the cable degradation analysis, both in the diagnostics and the prognostics phases. Simplified cable degradation statistics could have been used for the prognostics phase, but the degradation models described in Chapter 5 permit analyzing complex stochastic behaviors and their impact in comparison to simplified models.

At any given time t_0 the insulator degradation state can be estimated from past and/or present measurements (*diagnostics* phase). Based on physics-based models, this analysis is affected by errors in the system inputs (that is, past and present measurements), but also by errors in the degradation model, unless a direct measurement of the degradation is available.

After t_0, the prediction of the future cable degradation is affected by the errors in the estimation of the loads that will be applied to the cable, causing deviations in the thermal conditions of the cable, and the prediction is also affected by the result of the diagnostics phase that defines the condition of the component at the beginning of the prognosis phase.

The accelerated physics-based models adapted for Field Programmable Gate Array (FPGA) developed in (Chapter 5) have the cable load as system input. Therefore, any deviations from the predictions of the cable load can have an impact on the cable prognosis. This chapter applies stochastic models of the grid load that considers those uncertainties, and predicts the RUL of the cable that guarantees a given cable reliability. The results describe the impact of these uncertainties on the cable lifetime.

A deterministic analysis with a single simulation is not representative for the possible future stochastic behavior of the grid. And similarly, a single physics-based model analysis of the instantaneous degradation level cannot represent the uncertainty errors introduced by the past cable monitoring and model deviations. Therefore, it is necessary to perform enough simulations (generally thousands to millions), that are representative of the stochastic behavior by performing a Monte Carlo (MC) simulation.

The result of the process is the expectation of the degradation level of the cable through time, and its confidence margins. The confidence margins vary with the stochastic properties in each phase (either diagnostics or prognostics). Usually the uncertainty during the prognosis phase grows with time, because the varying grid conditions cause cumulative deviations, causing a growing uncertainty on the component condition.

The differences in the estimated degradation paths obtained by a deterministic model based on a single simulation and a the degradation path obtained by a MC model including these uncertainties causes a $\Delta\tau_{\text{RUL}}$ RUL estimation error in the deterministic model as shown in Figure 6.1.

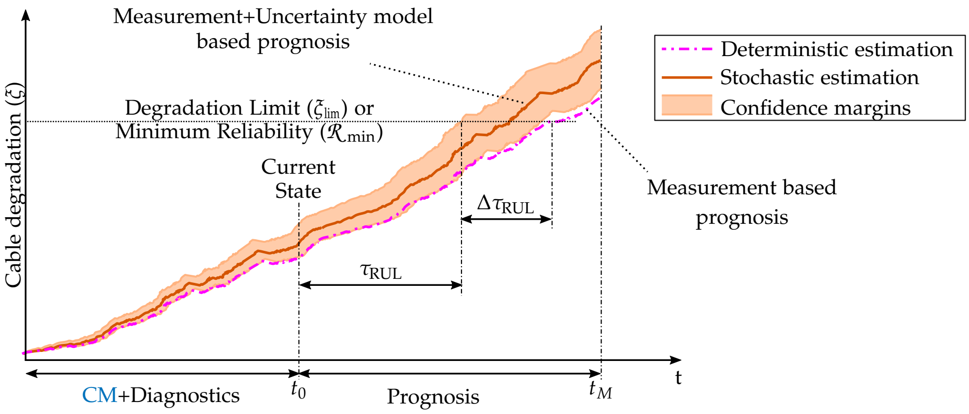


Figure 6.1: Effect of uncertainties on cable diagnostics and prognosis.

This chapter analyzes the impact of the different uncertainties on the RUL of the system, and test the validity of the method for producing more accurate RUL estimations.

6.1 IMPACT OF SYSTEM UNCERTAINTIES

The thermal behavior of underground cables is affected by several parameters that not only can have uncertain values, but the parameters are also be varying in time. The main factors affecting this behavior are (Alwan et al., 2016):

1. Load of the cable;
2. Varying soil temperature;
3. Varying soil thermal resistivity due to varying humidity (G. J. Anders, 2005, Chapter 4);
4. Variability in the conduit size;
5. Variability in the depth of burial and spacing between conductors;
6. Variability in cable dimension and materials;
7. Variability in the dimensions of the backfill;
8. Dry-zones that form around the cable due to the moisture migration.

Given the large number of uncertainties that effect the load of the cable, especially on the future load predictions, the focus of this thesis has been placed on the impact of the load variability. It would have been of great interest to cover the varying soil temperature, and the varying resistivity caused by the variable humidity. But due to the complexity of weather forecast statistics and soil humidity, the limited data available and time limitations, the thesis has concentrated on the effects of the load-related parameters.

The load-related uncertainties do not only have an impact on the prognostics phase, but also during the diagnostics phase. The accuracy of the load measurements has an impact on the temperature estimations of the cable, causing large deviations in the estimated instantaneous degradation levels (see Section 5.5.4).

A possible alternative for the diagnostics phase is using direct temperature measurements as inputs to the degradation models. The method certainly reduces the complexity on the estimation and reduces the uncertainty sources in this phase. However, the method requires replacing existing cables with cables equipped with fiber optics or alternative temperature sensors, which may not always be possible (Yilmaz and Karlik, 2006).

If direct temperature measurements are available, the process of diagnosing the instantaneous cable degradation is simplified, as the thermal model simulation becomes unnecessary for the diagnostics phase, and only the measurement deviations and the degradation models are to be applied. In this case, the simulation and analysis method for the prognosis phase remains the same.

6.1.1 *Impact of Measurement Uncertainties*

The instantaneous cable load can be measured employing different types of current transformers or Hall effect sensors (K.-L. Chen and N. Chen, 2011; Pecht, 2008), with the Rogowski coils being a common approach for transient measurements (Shafiq et al., 2014).

The various load sensors are subject to both calibration and noise errors. Calibration errors or sensor bias can generally be considered constant through the cable diagnostics process, while the noise is often modeled as a Gaussian stochastic variable.

However, the stochastic model must consider the variability between sensors, as the calibration can differ between any two manufactured sensors. Once a sensor is installed, the calibration bias will remain constant. But the calibration error of any given sensor installed for the measurements is uncertain. Accordingly, each MC trial has to consider the different possible sensors, where the calibration bias is distributed according to the statistical properties of the manufacturer calibration process.

The sensor noise, on the other hand, varies in time. A Gaussian parametric model is typical for representing the sensor noise. However, a Gaussian current value deviation does not cause a normally distributed error in the temperature estimation and the resulting cable degradation, due to the non-linearity of the models. This is true even when considering that the current measurement noise has negligible impact on the resistivity of the conductor and therefore on its resistance R.

Consider a load transient $\Delta I(t_0)$ that is applied at a given time t_0, causing a power transient ΔW_c, and the transient is measured with a current sensor that has an error modeled as $\varepsilon_I(t) \sim \mathcal{N}(\sigma_I, 0)$. Even if this deviation of the current measurements is considered to have negligible impact on the resistance of the main conductor, the estimated power transient $\Delta\hat{W}_c$ will be affected by an error $\varepsilon_{\Delta W_c}(t)$:

$$\Delta\hat{W}_c(t_0) = (\Delta I(t_0) + \varepsilon_I(t_0))^2 R = \Delta W_c(t_0) + \varepsilon_{\Delta W_c}(t_0) \tag{6.1}$$

therefore causing a power transient estimation error $\varepsilon_{\Delta W_c}(t_0)$ of

$$\varepsilon_{\Delta W_c}(t_0) = (2\Delta I(t_0)\varepsilon_I(t_0) + \varepsilon_I^2(t_0))R \tag{6.2}$$

that includes the sum of a normally distributed and χ^2 distributed terms.

The temperature transient caused by the actual dissipated power is described by the self-heating, mutual-heating the soil effect processes. Given that the temperature transient $\Delta\theta(t)$ caused by a constant power transient is proportional to the amplitude of the power transient according to Equation (3.38), a transient temperature-per-watt curve $C(t)$ can be defined for a given cable-set structure. Therefore, a power estimation error $\varepsilon_{\Delta W_c}(t_0)$ results in an estimated temperature transient $\Delta\hat{\theta}(t)$ with an error $\varepsilon_{\Delta\theta}(t)$

$$\Delta\hat{\theta}(t) = \Delta\theta(t) + \varepsilon_{\Delta\theta}(t) = (\Delta W_c(t_0) + \varepsilon_{\Delta W_c}(t_0))\, C(t) \tag{6.3}$$

$$\varepsilon_{\Delta\theta}(t) = \varepsilon_{\Delta W_c}(t_0) C(t) = (2\Delta I(t_0)\varepsilon_I(t_0) + \varepsilon_I^2(t_0))\, R\, C(t) \tag{6.4}$$

When multiple transients are applied, the superposition of the results from the consecutive load transients is to be applied to obtain the resulting temperature estimation error $\varepsilon_\theta(t)$.

The estimation error $\varepsilon_{\Delta\xi}(t)$ for the degradation transient $\Delta\xi(t)$ caused by the measurement error of a single power transient, for an interval of Δt can then be obtained from Equation (3.61). When applying the Zhurkov degradation model in Equation (3.58)

$$\varepsilon_{\Delta\xi}(t) = \frac{\Delta t}{\tau_0 \left(\frac{-\ln(1-P_{F,\mathcal{F}})}{\mathcal{D}}\right)^{1/\beta}} \left(\mathrm{e}^{-\frac{w_0 - \chi E}{R_G(\vartheta(t)+\varepsilon_\theta(t))}} - \mathrm{e}^{-\frac{w_0-\chi E}{R_G \vartheta(t)}} \right) \tag{6.5}$$

or

$$\varepsilon_{\Delta\xi}(t) = \frac{\Delta t}{\tau_0 \left(\frac{-\ln(1-P_{F,\mathcal{F}})}{\mathcal{D}}\right)^{1/\beta}} \left(\mathrm{e}^{-\frac{w_0 - \chi E}{R_G\left(\vartheta(t)+(2\Delta I(t_0)\varepsilon_I(t_0)+\varepsilon_I^2(t_0))\, R\, C(t)\right)}} - \mathrm{e}^{-\frac{w_0-\chi E}{R_G \vartheta(t)}} \right) \tag{6.6}$$

where $\varepsilon_I(t) \sim \mathcal{N}(\sigma_I, 0)$.

No analytical solution is found for the expected value of this expression. Obtaining a numerical solution from Equation (6.6) is possible via simulation. However, it is important to notice that the above development is only valid assuming that the thermal variation is small enough to consider the thermal resistivity variation to be negligible. Therefore, it cannot be applied to larger load deviation situations, which can be caused e.g. by grid switching load transients, or sensors with larger calibration errors.

Obtaining an expression for the degradation error for a varying load input is even more complex because the full convolution needs to be applied to account for the superposition of the thermal responses.

$$\hat{\theta}(t) = \theta(t) + \varepsilon_\theta(t) = (\Delta W_c(t) + \varepsilon_{\Delta W_c}(t)) * C(t) \tag{6.7}$$

$$\begin{aligned} \varepsilon_\theta(t) &= \varepsilon_{\Delta W_c}(t) * C(t) = \int_0^t C(t-\tau)\varepsilon_{\Delta W_c}(\tau)\mathrm{d}\tau \\ &= \int_0^t C(t-\tau)(2\Delta I(\tau)\varepsilon_I(\tau) + \varepsilon_I^2(\tau))\, R \end{aligned} \tag{6.8}$$

$$\text{being } \varepsilon_I(t) \sim \mathcal{N}(\sigma_I, 0) \tag{6.9}$$

which should be replaced in Equation (6.5).

The use of the full electro-thermal model accelerated in FPGA permits performing an accurate analysis of the errors caused by these current measurement deviations, considering the superposition for varying loads and the variation of the resistance of the conductor with the temperature.

6.1.2 *FPGA Architecture for Modeling Measurement Uncertainties*

This section describes the modeling for a use case where a sensor ℓ measures the load applied to a cable-set. The sensor is characterized by an offset calibration error o_ℓ that varies between sensors with standard deviation σ_o and offers a Gaussian measurement noise error $\eta(t)$ with standard deviation σ_η, both defined relative to the rated current of the cable I_0. That is, the total load current estimation error $\varepsilon_{I\ell}(t)$ for any sensor ℓ will be

$$\varepsilon_{I\ell}(t) = o_\ell + \eta(t) : o_\ell \sim \mathcal{N}(0, I_0\sigma_o), \eta(t) \sim \mathcal{N}(0, I_0\sigma_\eta) \tag{6.10}$$

The sensor offset calibration error o_ℓ is assumed to remain constant through the whole life of each sensor, while the noise error $\eta(t)$ is time dependent.

The sensor bias and noise models are implemented by the uncertainty generator block shown in Figure 6.2. The block also permits injecting uncertainties related to the prognostics phase. Therefore, the block is controlled by the Stochastic Activity Network (SAN) grid model, and capable to generate varying load factors f_{Load}, corresponding to each condition of the grid, causing stochastically varying loads.

The sensor calibration bias and noise are generated by two Gaussian Pseudo-Random Number Generation (PRNG) blocks. The PRNG blocks are based on a uniform PRNG and an implementation of the inverse Cumulative Distribution Function (CDF) algorithm (Gutierrez, Torres, and Valls, 2012). The pseudo-random values are multiplied by the sensor noise standard deviation and bias values that are inputs to the block and permit simulating the architecture with configurable sensor characteristics or time-varying sensor characteristics.

The sensor bias PRNG generates a new random number at the beginning of each MC trial, controlled by a `SIMStart` signal issued by the SAN scheduler. Therefore, its value is kept constant through the whole simulation, in order to simulate an individual sensor ℓ, which is different for each MC trial.

The outputs are connected to the pipelined power sequence generator block shown in Figure 5.29, that drives the electro-thermal model.

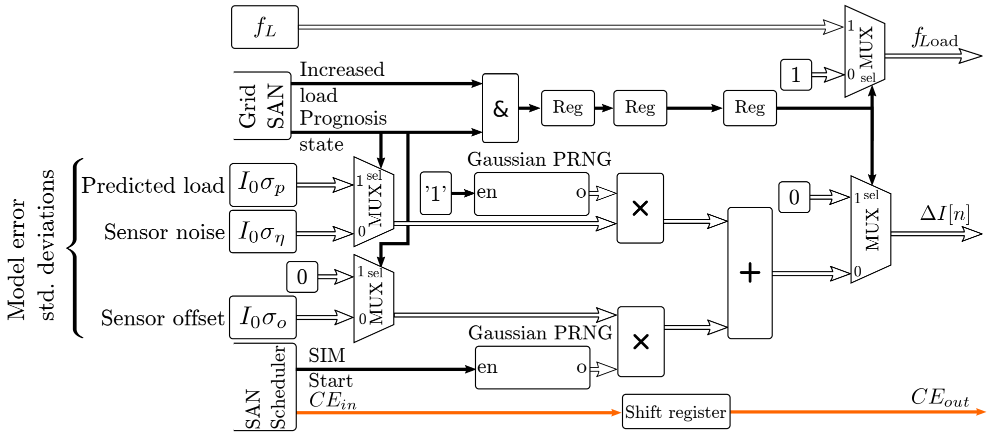


Figure 6.2: Pipelined load uncertainty model.

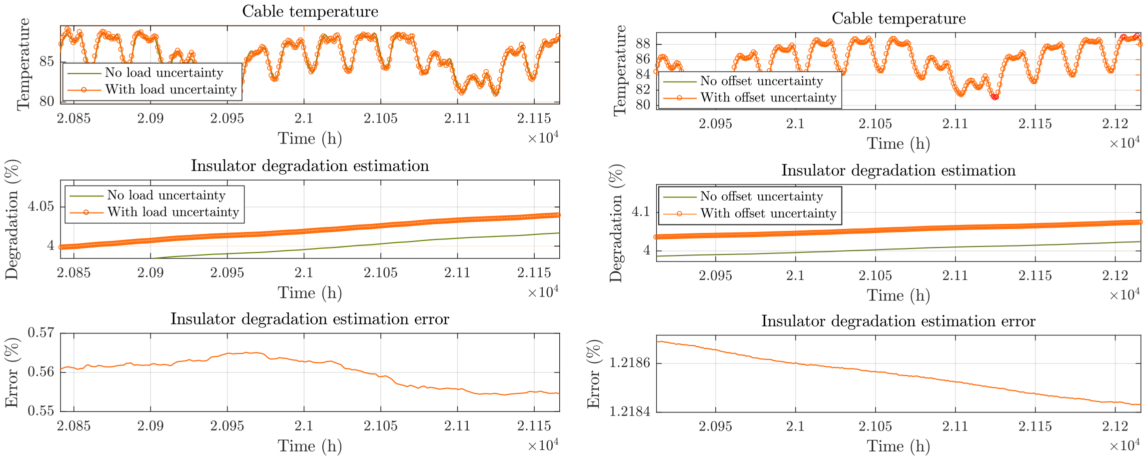
6.1.3 *FPGA-based Analysis of the Impact of the Measurement Uncertainties on The Electro-Thermal Degradation Model*

The degradation process of Cable A in Appendix A.1 was analyzed as use case. Three simulations were performed on the FPGA:

1. A single deterministic simulation with no sensor deviations considered;
2. A MC simulation with N_{MC}=100 × 10^3 trials, considering the sensor calibration bias uncertainty;

3. A MC simulation with N_{MC}=100 × 10^3 trials, considering the sensor noise uncertainty.

Figure 6.3a compares the simulation considering no measurement uncertainties to a single MC trial with Gaussian sensor noise with $\sigma_\eta = 1\%$ standard deviation relative to the rated current I_0. Figure 6.3b shows the effect of the load sensor calibration bias, with standard deviation $\sigma_o = 0.2\%$ relative to I_0.



(a) (b)

Figure 6.3: Effect on the degradation estimation by: (a) a 1% Gaussian standard deviation error on the load estimation, (b) a 0.2% load sensor bias .

It can be observed that, in comparison, a lower calibration offset error has a greater impact on the degradation estimation, compared to the sensor noise, due to the error being always biased in the same direction. That is, the cumulative effects of the Gaussian noise cancel partially, although not completely, due to the non-linearity of the degradation process (see Section 6.1.1). The results of a single trial also show that the impact of a very low value calibration bias on the current sensor can have a large impact on the predicted lifetime values of up to 1.21 %. Therefore, the errors caused by measurements may not be negligible for the reliability analysis.

The accuracy of the thermal responses must be pondered when interpreting these results. It should be noticed that the degradation level deviation caused by the sensor calibration bias is above the average expected model deviation from the designed accuracy levels from the degradation model (see Section 5.5.4).

Notice the graphs shown in Figure 6.3 correspond to a single MC trial, which may not be representative for the expected cable lifetime figures. In order to evaluate the expected deviation of the degradation, a full MC simulation is necessary, where the N_{MC} are representative of the actual sensor uncertainty distributions. Figure 6.4 shows the Probability Density Function (PDF) for the degradation level $\xi(t_0)$ at an arbitrary time $t = t_0 = 353\,000\,\mathrm{h}$, based on 100 × 10^3 MC simulations performed in the FPGA. The trials were simulated in the FPGA and the degradation levels at the given time captured as the marking of an extended place.

In the case of the Gaussian sensor noise, it can be observed that despite the symmetric distribution of the noise with null mean value, the deviation in the degradation levels is positive, due to a) the χ^2 distributed term in Equation (6.2) that causes the expectation of the dissipated power to be positive and b) the non-linear degradation function Equation (3.58) that causes shifting of the PDF towards a positive thermal degradation bias. Even if a temperature sensor were employed during the diagnostics phase, its normally

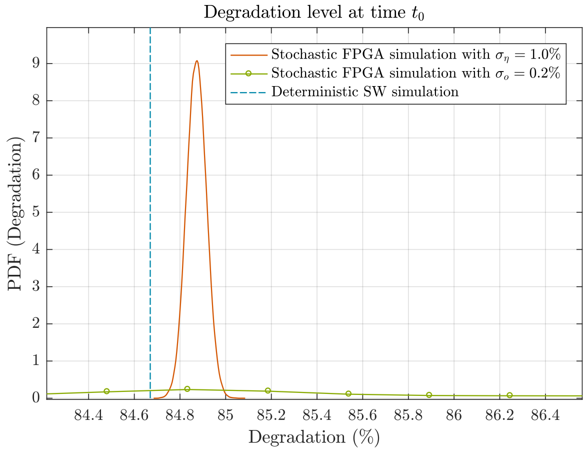


Figure 6.4: Degradation state PDF at the beginning of the prognostics stage $t = t_0$.

distributed noise would be transformed by the exponential degradation model in Equation (3.58) into a log-normal distribution, leading to a positive expected deviation of the cable degradation.

6.1.4 *Impact of Grid Switching Uncertainties on the Electro-Thermal DegradationProccessEstimation*

When the electrical power distribution grid switches randomly, the load applied to the cable can vary according to different load factors and statistics, depending on the new grid topology. The impact of this grid state on the RUL of the cable can be larger than the sensor measurement deviations, due to the higher variability in the dissipated power, and the variations caused on the thermal condition as a consequence. The statistical distribution of the switching and their duration can greatly affect the maximum temperatures reached by the cable in those conditions. However, their statistics are complex to solve analytically, due to the synergy between the components that causes probabilities that govern each component not to be independent. For example, when multiple cables are under analysis, the failure of a cable causes a grid to switch or change its condition, and the new grid state can alter the conditions under which the other cable is degrading.

The analysis of such systems can be performed based on the architecture described in Sections 2.4 and 4.5.1.6. The architecture solves the synergism between the grid and the cable, while taking advantage of a pipelined hardware architecture for simulating the physics-based degradation model described in Chapters 3 and 5.

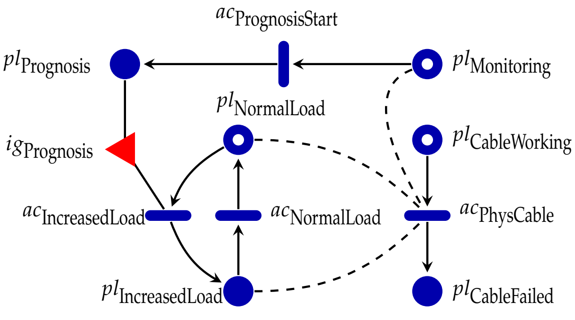
In order to evaluate the architecture for the prognosis of a grid, the following use case has been analyzed.

A cable-set with the characteristics of Cable A described in Appendix A.1 is installed as part of a grid where past measurements until $t = t_0 = 353\,000\,\text{h}$ are known, and the future yearly load patterns are predicted. The simulation for the center cable conductor temperature is performed, and the Zhurkov electro-thermal degradation model is used to estimate the degradation of the insulation under those temperature conditions. The polymer degradation parameters for Cross-Linked Polyethylene (XLPE) have been obtained from (Mazzanti, 2009). Realistic yearly load data values for the cable were obtained Red Eléctrica de España (Red Eléctrica de España, 2019), and data was scaled to the rating of the cable for full-year hourly currents ($\tau_s = 1\,\text{h}$). The load data were recorded into the FPGA load generator Block RAM (BRAM) described in Section 5.5.5.

The model of the grid predicts that randomly, due to the abnormal switching of the grid, the load factor of the cable will increase from a normal load factor f_{Load}= 1.0 to

$f_{\text{Load}} = 1.2$ (a 20% increase in comparison to normal load cycles). These events are modeled to be exponentially distributed with an occurrence rate of $\lambda = 2.31 \times 10^{-4}\,\text{h}^{-1}$ (an average period of 6 months), and of each event lasts for $\tau_{\text{Inc}} = 6\,\text{h}$ (deterministic).

The SAN model for this grid is shown in Figure 6.5. When the cable is in normal conditions, place $pl_{\text{NormalLoad}}$ is marked, and $\text{m}(pl_{\text{IncreasedLoad}}) = 1$ when the grid is under an abnormal switching condition.



Activity	Distr.	Param.
$ac_{\text{PrognosisStart}}$	Determin.	$\tau = t_0$
$ac_{\text{IncreasedLoad}}$	Exponential	$\lambda = 2.31 \times 10^{-4}\,\text{h}^{-1}$
$ac_{\text{NormalLoad}}$	Determin.	$\tau = 6\,\text{h}$
$ac_{\text{PhysCable}}$	Physics	

Gate	$\text{Pred}_{ig}()$
$ig_{\text{Prognosis}}$	$\text{m}(pl_{\text{Prognosis}}) \geq 1$

Gate	$f_g()$
$ig_{\text{Prognosis}}$	-

Figure 6.5: Stochastically switching grid load model implemented for validation purposes, including the switching from monitoring to prognostics state.

The purpose of the architecture is to perform both the diagnosis of the current condition at a given time t_0, and the prognosis of the cable after that time. Therefore, the past cable simulation also needs to be performed in order to evaluate the current condition. During the monitoring of the cable, the load factors applied to the cable are known, and no stochastic grid switching (no random increased load factors) needs to be considered, but the load measurement uncertainties must be taken into account.

Therefore, the stochastic properties that govern the cable vary between the monitoring and prognosis phases. This parameter switching can be simply integrated within the SAN model itself, by introducing a deterministic interval activity with interval value t_0 that switches between two states: $pl_{\text{Monitoring}}$ and $pl_{\text{Prognosis}}$.

The marking of $pl_{\text{Prognosis}}$ enables activity $ac_{\text{IncreasedLoad}}$ via input gate $ig_{\text{Prognosis}}$ and initiates the prognosis cycle with stochastic load factors applied to the cable.

The physics-based electro-thermal degradation model for the cable-set is implemented in $ac_{\text{PhysCable}}$. The activity completes when the degradation level of the cable reaches the given $\xi_{\text{lim}} = 1.0$, switching the marking from $pl_{\text{CableWorking}}$ to $pl_{\text{CableFailed}}$.

The load factors applied to the cable vary depending on the grid state. The marking of the places $pl_{\text{NormalLoad}}$ and $pl_{\text{IncreasedLoad}}$ are used to drive the load factor generator in Figure 6.2 that accordingly drives the cable load generator in Figure 5.29. The PDF distribution of activity $ac_{\text{PhysCable}}$ depends on the marking of those places, which has been indicated by dashed lines.

Initially the cable is set to be in monitoring state and normal load. The initial marking for the model has been indicated in the figure as tokens for readability purposes.

In order to validate the simulation architecture, the SAN model in Figure 6.5 including the cable-degradation simulator described in Chapter 5, and the SAN simulator architecture in Figure 4.25 that is described in Chapter 4.

Figure 6.6 shows a capture of the temperature results from the pipelined electro-thermal cable model of the cable, when it is being impacted by the SAN grid model simulator. In order to simplify the validation process of the model, the extension of the increased load factors events was extended to 24 h intervals for this specific capture. In this sequence the grid is initially in the *Normal load* state. Before the state change of the grid model to the increased load factor, the electro-thermal model is ① *flushed*, that is, the pipeline continues processing until the last valid input is processed, while all inputs are flagged as invalid. At this point, the thermal model reports the current cable degradation state and the pipeline is paused until the ② new state is defined. Notice that the cable degradation state can alter the behavior of the grid model. The pipeline is then re-enabled and continues filling the pipelines of the ③ load calculation and the ④ thermal model before the output is valid. The same process repeats at (1′) when a new state is generated, such as the return to the normal load factor state.

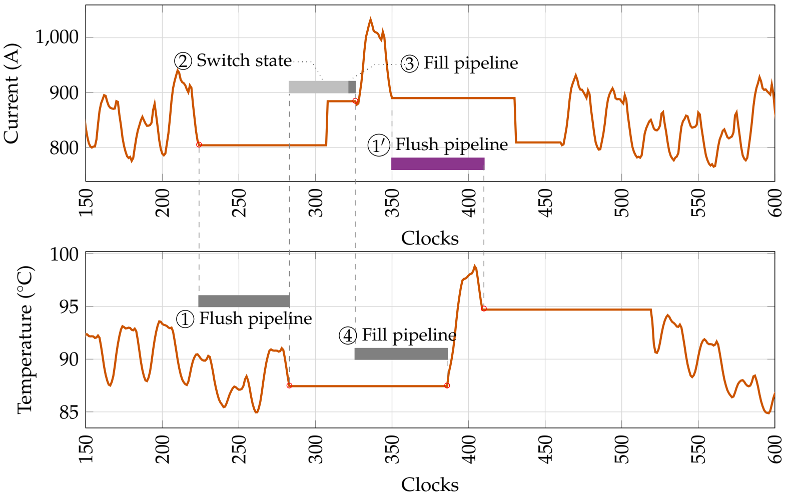


Figure 6.6: Behavior of the thermal model pipeline during a grid state change causing a load increase.

Figure 6.7 shows the resulting impact on the cable temperature caused by the load factor increase in Figure 6.6 .

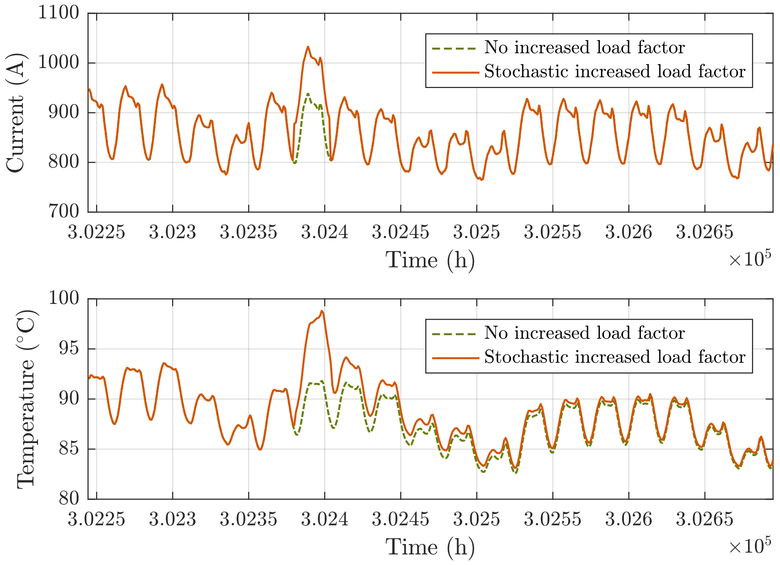


Figure 6.7: Effect of stochastic load factor increases during the prognostics stage.

It can be observed that each random switching in the grid can cause large temperature variations that extend for periods much longer than the duration of the events. These transitions can cause large deviations from the results obtained from deterministic yearly load patterns. Their impact on the system RUL is reduced by their low probability of occurrence that is considered in the reliability estimation results.

6.1.5 *Synthesis and Resource Usage of the FPGA architecture*

An FPGA simulator for the SAN model in Figure 6.5 was designed in Möbius Tool (*The Möbius Tool* 2017) software and the generated design was converted to a Xilinx Vivado® block design (Garro et al., 2019b). The parameters of Cable A and load patterns were imported into the FPGA design, and synthesized for a Xilinx XCZU9EG MPSoC with speed grade -2 at a 187 MHz Programmable Logic (PL) fabric clock rate.

The FPGA design utilization, clock speed and the timings from the synthesis results are displayed in Table 6.1 and Figure 6.8 shows the generated FPGA layout. In this case, the parallelization of multiple simulators was not possible due to the resource usage requirements of the electro-thermal model simulator. It can be observed that, due to the structure of the mutual-heating and soil effect filters described in Section 5.5.3, most of the Configurable Logic Blocks (CLBs) of the FPGA are dedicated to the shift registers of the secondary filter stages, which save resources for the arithmetic computation. No BRAM have been assigned for this purpose, due to technological limitations in the used FPGA that do not permit randomly resetting the internal register cells. In this implementation, the mutual heating process and soil effect filters were implemented as separate filters, that permitted the debugging and validation of the implementation. However, both filters can be merged can be merged into a single filter, saving large resources and reducing incremental errors as a consequence of the addition of both processes.

Table 6.1: FPGA resource utilization

FPGA Device	(XCZU9EG-2)			
Time and thermal values	**Value**			
PL Clock	187 MHz			
Worst Negative Slack (WNS)	0.090 ns			
Worst Hold Slack (WHS)	0.010 ns			
Worst Pulse Width Slack (WPWS)	3.750 ns			
Thermal Margin@25.0 °C	68.4 °C			
Resource	**Usage**	**% of Total**	**FIR Mutual**	**FIR Soil**
LUT	152830	55.76%	47766	72359
Registers	114947	20.97%	48451	37303
BRAM	35	3.84%	0	0
DSP	1663	65.99%	1374	273

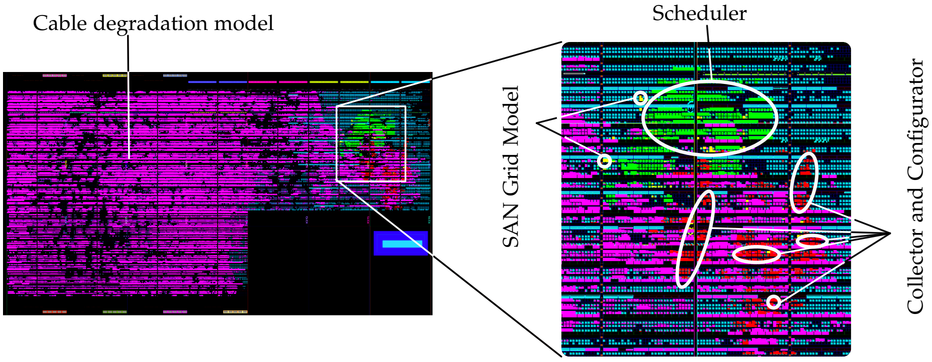


Figure 6.8: Layout of the implemented FPGA design.

6.1.6 *Simulation Time and Resource Usage Comparison with Parallel Software Solutions*

An equivalent Software (SW) model was designed in Matlab®R2017b for model validation purposes. However, Matlab based solutions are generally not optimal for the simulation of sequential processes. The model, run without considering uncertainties was found to require 270 s to perform a single simulation up to the cable failure time, on a 12 core Intel®Xeon®E5-2690 v3 @2.60 GHz, with 128 GiB RAM. However, the memory resource usage was found to be unacceptable for the purpose of online reliability analysis on embedded equipment that generally lack those resources, and the simulation time required to perform the thousands of MC trials would be impractical.

In order to evaluate the efficiency of the FPGA architectures, a Möbius Tool simulation (*The Möbius Tool* 2017) was adapted to be accelerated for Graphics Processing Unit (GPU) simulation. The parallelization of SW on multi-cpu and/or GPU architectures for the acceleration of simulations is common, but it has been shown that careful planning of memory resources is necessary for an efficient use of the architectures (Yianni et al., 2018).

In a SAN grid model simulation, the software implementation must include two separate parts:

1. The SAN model simulator, including the activity sequencing and the necessary PRNG;
2. The cable degradation functions implemented within the $ac_{\text{PhysCable}}$ activity, that describe the activity completion interval.

GPUs are especially suited to accelerate signal processing calculations that require thousands independent parallel operations, better than complex sequential and non-independent sequences involved in the SAN model. Therefore, they are especially suited for the acceleration of matrix operations like those involved in the convolution processes of the electro-thermal model. In this implementation, the electro-thermal model was adapted to a C++ code to be run accelerated on a Nvidia CUDA platform (NVIDIA Corporation, 2019), using version 3.6.2 of ArrayFire open-source library (AccelerEyes, 2019).

The sequential software part that simulates the SAN model can be auto-generated by Möbius Tool. Unfortunately, while the SAN formalism does not prevent that an activity completion interval depends on the instantaneous state of the places (their marking) and be conditioned by the completion time intervals of the rest of activities (and therefore to model physics-based activities), the Möbius tool does not permit such implementations.

In order to adapt the model for a Möbius Tool simulation, the physics-based activity can be transformed as in Figure 6.9.

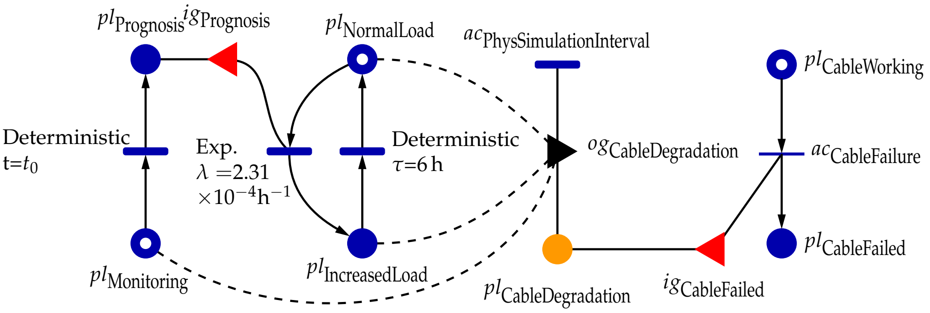


Figure 6.9: Adaptation of the SAN model in Figure 6.5 for execution in Möbius Tool.

In this adapted model, the execution of an output gate $og_{\text{CableDegradation}}$ implements the degradation model of the cable, which is evaluated at 1 h intervals and accelerated by the

GPU. The execution interval is dictated by a deterministic interval activity $ac_{\text{PhysSimulationInterval}}$. The degradation level is written in the extended place $pl_{\text{CableDegradation}}$, which is evaluated by input gate $ig_{\text{CableFailed}}$, that enables or disables the completion of an instantaneous activity $ac_{\text{CableFailure}}$.

For this simulation, both in Möbius Tool and the FPGA simulation, the cable was modeled to be initially As Good as New (AGAN) (that is, with an initial degradation of $\xi(0) = 0\%$) and subjected to yearly load patterns obtained from (Red Eléctrica de España, 2019). The beginning of the prognostics stage was configured to be $t = t_0 = 353\,000\,\text{h}$, and the simulation was run up to the cable failure time or a mission time of 80 years, whichever first occurred. 1×10^5 trials were performed to obtain acceptable confidence margins.

The final SW simulation was developed using Möbius Tool version 2.5 that permits multi-core Central Processing Unit (CPU) parallelization, and the electro-thermal model was accelerated employing a GPU. The software was run parallelized on 12 cores of an Intel® Xeon® E5-2690 v3 @ 2.60 GHz, with 128 GiB Random Access Memory (RAM) and a Nvidia Quadro®K4200 with GPU with 1344 cores and 4 GiB memory.

In the SW test, 12 MC simulation trials were parallelized onto the 12 available CPUs. The simulations required 2315 s to finish. While the results were faster than the initial simulations the performance does not suffice to execute the 10^5 MC simulations required for the uncertainty analysis.

In order to profile the FPGA simulation architecture timings and performance, in-FPGA cores were added to the architecture that counted execution clock cycles for Direct Memory Access (DMA) transfers, activity completions and the electro-thermal simulation behavior, without these measurements being affected by the latencies caused by interrupts, buses, or context switching. As displayed in Table 6.2, each single FPGA simulation took 2.3 ms to execute, performing at 434 simulations/s, 71×10^3 times faster than the software solution accelerated by GPU.

At a clock rate of 187 MHz, the electro-thermal model alone was expected to require 222 s to execute the simulation up to the expected cable lifetime. Therefore SAN model simulator architecture had only a low 3 % impact on the total simulation time.

This speed improvement permits the execution of thousands of MC simulations to analyze the impact of the uncertainties in the model.

Additionally, the FPGA architecture mechanism for the reward capture method described in Section 4.7 was found to be much more efficient in terms of memory usage. Out of 46×10^7 activity completions, only 0.2×10^6 state changes required to be captured. The software simulations required 9.18 GiB RAM memory, generally unavailable on low-cost and on-site embedded equipment, while the FPGA simulation required 549 times less memory space for transferring the results.

Table 6.2: Resulting simulation timings and memory usage

Monit. places	Output Resolution (h)	Completions	Captured changes ($\times 10^6$)	**Simulation time and memory resources**						
				SW		**FPGA**				
				CPU (s)	RAM usage (GiB)	HW sim. time (s)	DMA time/size (s) / (MiB)	Total time (s)	Speed gain	RAM gain
1	24	46 $\times 10^7$	0.2	1.92 $\times 10^7$	9.18	230.48	46 10^{-3} / 1.53	270.01	71 100	549

6.2 Impact of the uncertainties on the predicted RUL

In terms of high-reliability systems, the RUL of the system is limited by the acceptable reliability level. Once the equipment reliability is below this given limit $\mathcal{R}_{\text{min}}$, the equipment is considered not useful, or not ready to be used. Therefore, the prognosis of the cable must predict when then cable reliability will reach this limit, in order to predict how much life is left for the cable, under the given grid model. In order to evaluate the cable failure reliability, the marking of place $pl_{\text{CableFailed}}$ in Figure 6.5 was captured by the result collector, to be used as reward.

Using the CPU interface, varying parameters were introduced to the PRNG in the uncertainty generator in Figure 6.2 to generate different sensor uncertainty characteristics, including sensor noise and calibration bias. And similarly, varying rates were configured for the grid switching. 100 000 trials were performed for each MC, which records the captured results in RAM.

The CPU was then used to obtain the PDF for the place $pl_{\text{CableFailed}}$, and to calculate mean values and variances out of the MC simulation. It is important to note that in Figure 6.5 this place does not denote the cable failure event, but rather that the cable has failed in the past and it remains in failed state. Therefore, the PDF of the place marking actually refers to the cable failure CDF.

Therefore, the reliability of the cable under those conditions can be evaluated by

$$\mathcal{R}(t) = 1 - \int_0^t f_{CableFailure}(t)\mathrm{d}t = 1 - F_{CableFailure}(t) = 1 - f_{pl_{\text{CableFailed}}}(t) \tag{6.11}$$

Figure 6.10 shows the MC simulation result for a case in which the prognosis has been performed at time $t_0 = 353\,000\,\text{h}$. The monitoring of the cable load has been performed with a sensor with $\sigma_\eta = 1.0\%$ and a variability in offset calibration error among sensors of $\sigma_o = 0.25\%$ relative to the rated current. For the prognosis stage, the predicted future load is also considered uncertain, with a normally distributed error with standard deviation of $\sigma_p = 5\,\%$ of the rated current.

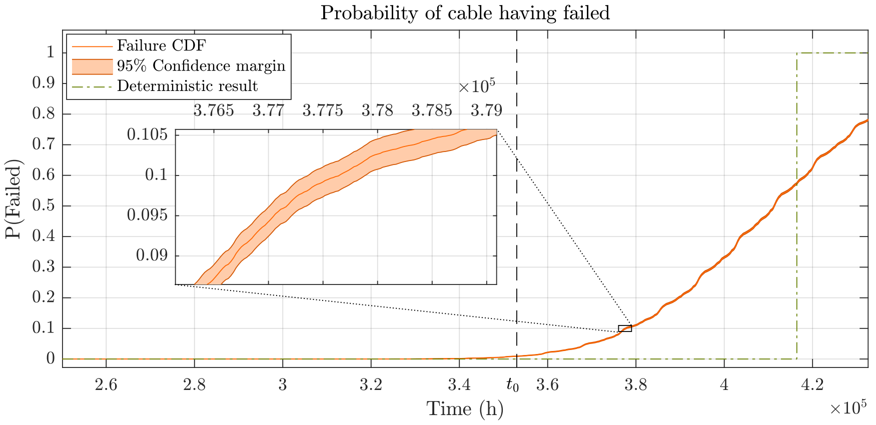


Figure 6.10: FPGA based results considering uncertainties: $\sigma_\eta = 1.0\,\%$, $\sigma_o = 0.25\,\%$, $\sigma_p = 5\,\%$, $\lambda = 2.31 \times 10^{-4}\,\text{h}^{-1}$, and $\tau_{\text{Inc}} = 6\,\text{h}$.

The software simulation results without uncertainties predicted a cable RUL of 63 388 h. When both diagnostics and prognosis stage uncertainties are considered with 100 000 simulations in the FPGA, the results show that there is a 10 % probability that the insulator will have failed $\Delta\tau_{\text{RUL}} = 38\,748\,\text{h}$ earlier (90% reliability time). At a 99% cable reliability requirement, the RUL of the cable is reduced by $\Delta\tau_{\text{RUL}} = 62\,556\,\text{h}$.

In order to analyze the impact of load prediction uncertainties in the prognostics phase, a sensitivity analysis was performed by running the FPGA model with varying standard deviations σ_p between 0% and 9%, relative to the rated current $I_0 = 1065\,\mathrm{A}$. For $\sigma_p = 2\%$, the FPGA simulation estimated a RUL of 61 896 h, 0.8 % earlier than the software simulation, which can be acceptable for this application. However, when the prognostics uncertainty grows, the RUL estimation error was found to increase quasi-geometrically with the future load uncertainty, as shown in Figure 6.11. The results are compared to the case when stochastic grid switching with a load factor of $f_L = 1.2\%$ is considered in the prognostics phase.

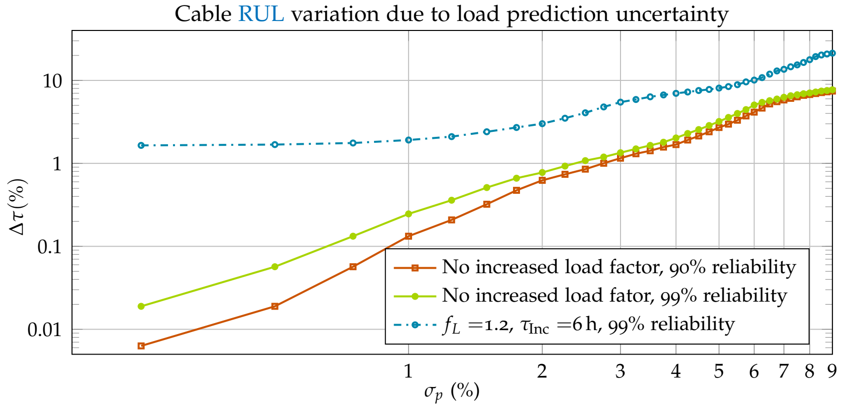


Figure 6.11: Expected RUL error due to load uncertainty during prognostics stage, relative to the predicted RUL.

The stochastic switching of the grid is shown to cause further degradation. The impact of the switching varies depending on the season of the year, because the different loads cause varying cable temperatures, and the derivative of the degradation equation Equation (3.58) is temperature dependent.

Finally, when the model shown in Figure 6.5 is considered for the load increase periods in the prognostics stage, the expected RUL drops between 1.6 % and 14.7 %.

6.3 ACCURACY OF THE OBTAINED RESULTS

The accuracy of the results in Figure 6.11 is limited by the accuracy of the thermal heating model approximations described in Section 5.5. When producing the mutual-heating and soil effect process models, the temperature curves were designed to have a 0.2 % accuracy.

It is important to consider this, because from Figure 5.28, it can be observed that the degradation values can be up to 2% higher degradation in comparison to actual values when the cable temperatures are close to 90 °C. This happens because of the high sensitivity of the Zhurkov degradation curves to temperature. However, subsequent positive and negative valued transient amplitudes cancel the effects of each other, preventing a cumulative error in the temperature and degradation values.

The accuracy can also be largely be improved by merging the two curves of the mutual heating process and the soil heating effect, to prevent cumulative errors.

While the impact of the sensor noise and bias on the estimation of the RUL was found to be relatively low (see Figure 6.11), the stochastic grid switching was found to have a meaningful impact, and the sensor uncertainties can cause additional estimation errors. This impact cannot be considered negligible, and the results are meaningful, even considering the complexity to obtain a high accuracy by the electro-thermal degradation model.

If an improved accuracy on the results is important, the filter segmentation with constant coefficients for the mutual-heating process and the soil effect described in Equation (5.67) can be approached also using linear regressions. If the filter is segmented as

$$\begin{aligned}&\sum_{i=M_\theta}^{L_\theta-1} \Delta W[n-i]\theta_v[i] \\ &\quad = \sum_{i=M_\theta}^{P_1-1} \Delta W[n-i]\theta_v[i] + \sum_{i=P_1}^{P_2-1} \Delta W[n-i]\theta_v[i] + \cdots + \sum_{i=P_{N-1}}^{L_\theta-1} \Delta W[n-i]\theta_v[i]\end{aligned} \tag{6.12}$$

where $P_1, P_2 \ldots, P_N$ are the sample time indexes where the filter is split, each segment j can be approximated by the linear regression:

$$\theta_v[i] \approx \beta_{0,j} + \beta_{1,j}(i - P_{j-1})\, \forall i \in P_{j-1} \ldots P_j - 1 \tag{6.13}$$

The output $y_j[n]$ of the j^{th} filter segment can be rewritten as an Infinite Impulse Response (IIR) filter (see Appendix A.6)

$$\begin{aligned}y_j[n] = {}& y_j[n-1] + \left(\beta_{0,j} + \beta_{1,j}\left(P_{j-1} - 1\right) - \beta_{1,j}P_{j-1}\right) \Delta W[n - P_{j-1} - 2] \\ &- \left(\beta_{0,j} + \beta_{1,j}P_j + \beta_{1,j}P_{j-1}\right) \Delta W[n - P_j - 1] \\ &+ \beta_{1,j} \sum_{i=P_{j-1}-1}^{P_j-1} \Delta W[n-i-1]\end{aligned} \tag{6.14}$$

6.4 Possible Adaptation Methods of Other Thermal Factors to the Cable Model

If the cable thermal models are not individually validated before their placement, or the variability of the cable laying dimensions is important, several uncertainty factors should be additionally included within the model for more accurate analysis of the thermal behavior (Alwan et al., 2016):

1. Enclosure of the cable
2. Installation of groups of cables
3. Depth of burial.
4. Thermal resistivity of the soil.
5. Conduit size.
6. Spacing between the cables corresponding to the different phases.
7. Varying dimensions of the backfill.

However, these uncertainties impact directly on the thermal step responses used for the simulation. Every time these factors are changed, the architecture must, therefore, be altered before each MC trial to evaluate this uncertainty condition. Applying this to the proposed architectures impacts directly on the filter weight values, but altering those live within the FPGA can be complex. Including variable weights in the filter multipliers would increase the use of multiplexers or require interfacing to a BRAM structure that would define the weights, probably causing very complex routing effects.

Future research lines may consider the impact of these parameters and forms to alter response curves for these varying factors, either by chained transformations or filter architecture redesigns.

Additionally, the impact of varying ambient conditions has not been evaluated through this thesis. The impact of the following factors would be of special interest in future research lines:

1. Varying cable size due to moisture migration.
2. Thermal resistivity and diffusivity variation due to the varying humidity of the soil.
3. Seasonal variability of the soil temperature.

Their impact causes the thermal models to be non-stationary, altering the step-responses. Specifically, the variability of the thermal resistivity due to humidity is known to vary by 3 times (G. J. Anders, 2005), which can greatly affect the estimated thermal behavior under a given load patterns. A correct evaluation of the process may require complex Finite Element Method (FEM) simulations, or methods to obtain approximate solutions if the variability of the parameters can considered quasi-stationary for certain periods.

7

CONCLUSIONS AND SUGGESTED FUTURE RESEARCH LINES

The main objective of the thesis was the study of Field Programmable Gate Array (FPGA) architectures for the reliability analysis and prognosis of high and medium voltage underground cables.

However, the reliability and the Remaining Useful Life (RUL) of those cables depends on the random behavior of the electric power distribution grid where the cable is placed. Electric power distribution grids are complex systems composed of a large number of components with complex degradation mechanisms. The parameters that affect the operation and failure of those components are often uncertain, and they require the use of statistical models, computationally solved by performing millions of simulations.

Existing software solutions did not perform the analysis of the models within reasonable time-frames, even using Graphics Processing Unit (GPU)-based architectures for their parallel simulation. The solutions also demand large memory resources that prohibit their use in on-site embedded equipment.

A Stochastic Activity Network (SAN) framework for the simulation of such grids and an FPGA architecture for its accelerated simulation have been developed in this thesis.

7.1 Discussion of Results

Using the described framework, the size and complexity of the grid models is reduced. Component models can be designed independent of the grid model, thanks to a unified integration of the physics-based degradation models within individual SAN activities.

The framework permits an efficient simulation of physics-based degradation models within a SAN grid model. The solution permits the parallel integration and simulation of physics-based models, where the simulation is run and paused on-demand, and it does not need to observe the grid simulation during its execution. This is advantageous to integrate efficient pipelined architectures in FPGA. The methodology has been validated by applying it to the analysis of underground cable-sets in a grid.

Physics-based degradation mechanisms of underground cables have been researched in depth, but they were found to be complex to be adapted to an FPGA simulation. Therefore, an adapted electro-thermal cable degradation model has been proposed that permits a highly parallel and pipelined FPGA-based simulation.

The FPGA framework was tested and validated by comparing it to existing Software (SW) simulations which were run parallelized in multi-Central Processing Unit (CPU) and GPU architectures. It was found that the FPGA solution offers the following benefits:

- A reduced cost equipment can be installed on-site for a unified reliability analysis and prognosis of an underground cable set.
- The simulations can integrate both grid and cable model uncertainties.
- The reliability and prognosis results are obtained within acceptable time-frames, in a fraction of the time required by SW solutions.
- The solution performs the analysis entirely in FPGA logic, freeing the CPU for other tasks.

- Memory requirements of the equipment are low, in comparison to SW solutions.
- The FPGA resource usage of the architecture grows linearly with the size of the models and it has a relatively low impact on the total simulation time.
- The architecture can be generated and synthesized automatically from SW based models.
- The synthesis method permits the generation and validation of the models by non-experts in the field of FPGA.

The FPGA architecture and framework discussed throughout this thesis was validated to perform an accurate electro-thermal degradation analysis of underground cables, at much lower resource usage and simulation times, compared to classical multi-CPU and GPU architectures.

Unlike traditional approaches that employ deterministic or reduced models, the presented architecture permits a complete and accurate physics-based model simulation that also includes the uncertainties existing in the grid models. The uncertainties were considered not only during the prognostics phase, but also during the monitoring of the cable. Small load measurement errors were found to have a large impact on the estimated cable degradation levels.

Although multiple sources of uncertainty exist in grid models, this research concentrated in the effect of the estimated load deviations on the cable reliability and remaining useful life. An impact analysis obtained from the simulations showed that these uncertainties cannot be neglected when estimating the remaining cable life, especially during the prognosis phase.

The framework permits the Hardware (HW) parallelization of multiple SAN model simulations. However, the synthesized electro-thermal cable degradation models are large, and therefore they limit the size of the grids that can be simulated in current FPGAs. This limitation prevents the simulation of multiple cable sets and performing multiple parallel simulations of the grid. Alternatives to further reduce the dimensions of the degradation models have been analyzed, although an actual implementation has not yet been performed.

The proposed simulation framework was based around the notion of linear time-invariant thermal models. However, some model uncertainties such as cable laying dimensions or ambient conditions were found to dynamically alter the system impulse response. Further research will need to be produced to be capable of evaluating the impact of those uncertainties.

7.2 Future Research Lines

Size constraints in the FPGA resources and time-varying component degradation models are limiting factors for the applicability of the method to real full-sized grid analysis.

Certain authors have approached solving complex electromagnetic transients with multi-FPGA architectures (Dinavahi and Y. Chen, 2013). Such an architecture may be capable of handling a full-sized grid for its statistical analysis with current FPGA dimensions. However, the impact of the latencies on the simulation times is still to be analyzed.

The simulation of time-varying models in this framework is mostly limited by the rigidness of the architecture to evolve to different structures within a single Monte Carlo (MC) trial or in between MC trials. Approximations such as varying time constants or modulating the responses could be possible, but further research would be required. Some authors have analyzed cable thermal processes through approximate circuits that avoid the complexity of non-linear exponential integral equations (Diaz-Aguilo and Leon, 2015; Diaz-Aguiló and León, 2015) that included time-varying parameters, which could provide a possible solution.

In comparison to full-sized cable and environment models, differential models of the electro-thermal models are inherently simple and they would simplify the simulation of time-varying processes. A Finite Element Method (FEM) solution for the electro-thermal model of the cable may therefore be ideal, but its adaptation to a fully parallel simulation can be challenging and the dimensions of the simulation may limit its applicability with current FPGA resources (Makhkamova, 2011).

A full realistic grid simulation model would also require further knowledge about the degradation mechanisms of other grid components such as string insulators or Circuit Breaker (CB), which are still being actively researched. However, if the statistical behavior of those components can be accurately modeled, the described framework opens new ways of performing the reliability analysis of a grid.

To conclude, this research has shown that standalone FPGA simulators are a valid and efficient solution for the reliability analysis and prognosis of underground cables, and that they can provide accurate reliability and RUL results through physics-based cable degradation models, in comparison to traditional deterministic methods. However, the accurate reliability and prognostics analysis of the complete grid remains challenging.

Part I

APPENDIX

APPENDIX 0A

A.1 Sample Cable-Set A

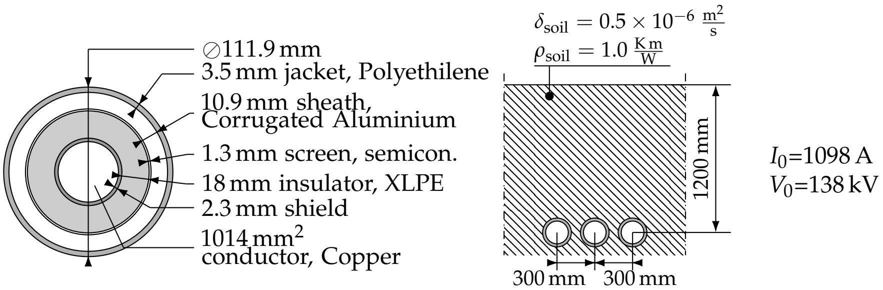


Figure A.1: Cable setup A, formed of 3 single-phase cables (American Furukawa Inc., 2018).

A.2 Sample Cable-Set B

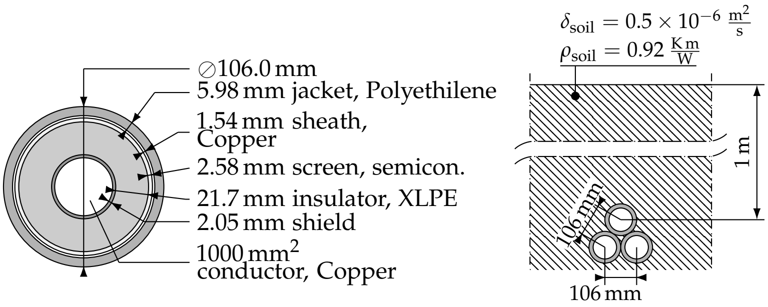


Figure A.2: Cable setup B (Diaz-Aguiló and León, 2015).

A.3 Implemented Field Programmable Gate Array (FPGA) blocks, interfaces and timing diagrams

Figure A.3 shows a block diagram of the signal interfaces, bus widths with their direction, and connectivity implemented. Figure A.4 describes the purpose of each bus. The example shows the design with a single activity, with one input gate and one output gate associated to this activity. Figure A.5 shows a flowchart diagram for the scheduler, and Figures A.6 and A.7 show the signal sequencing for the enablement and completion and acitivity between the scheduler and the activity and between the activity and the gates, respectively. The diagrams also describe the sequence in the case of a possible cancelation of the activity completion.

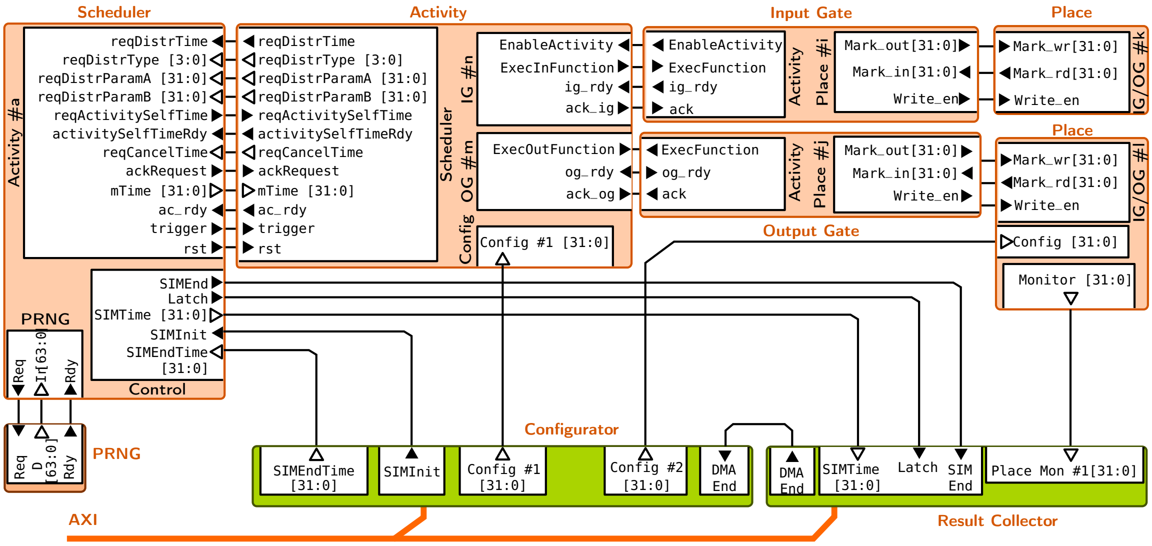


Figure A.3: Hardware (HW) block diagram of the FPGA Stochastic Activity Network (SAN) simulator with single activity, with one input gate and one output gate associated.

Scheduler-Activity

Signal / Bus	**I/O**	**Description**
reqDistrTime	I	Activity was enabled and requests new time to be scheduled
reqDistrType[3:0]	I	Distribution type requested by activity
reqDistrParamA[31:0]	I	Distribution parameter A
reqDistrParamB[31:0]	I	Distribution parameter B
reqActivitySelfTime	O	Request self-generated interval to the physics-based activity
reqCancelTime	I	Activity was disabled and requests canceling scheduled time
trigger	O	Request to trigger (complete) the activity
ac_rdy	I	Activity completed after trigger
ackRequest	O	Acknowledge requests from activity
rst	O	Reset activity state

Scheduler-Pseudo-Random Number Generation (PRNG)

Signal / Bus	**I/O**	**Description**
Req	O	Request new PRNG value
In[63:0]	I	Input PRNG value
Rdy	I	Input PRNG value is ready

OutputGate-Activity

Signal / Bus	**I/O**	**Description**
ExecOutFunction	I	Request to execute the associated function
og_rdy	O	Output function complete

InputGate-Activity

Signal / Bus	**I/O**	**Description**
EnableActivity	O	Predicate is met
ExecInFunction	I	Request to execute the function
ig_rdy	O	Input function complete

Input/Output Gate-Place(*PlaceRWi*)

Signal / Bus	**I/O**	**Description**
Mark$i$_out[31:0]	O	New marking for i^{th} place
Mark$i$_in[31:0]	I	Current marking of i^{th}place
Write_en$i$	O	Enable write of i^{th}place

Figure A.4: Interfaces between the generated FPGA SAN blocks.

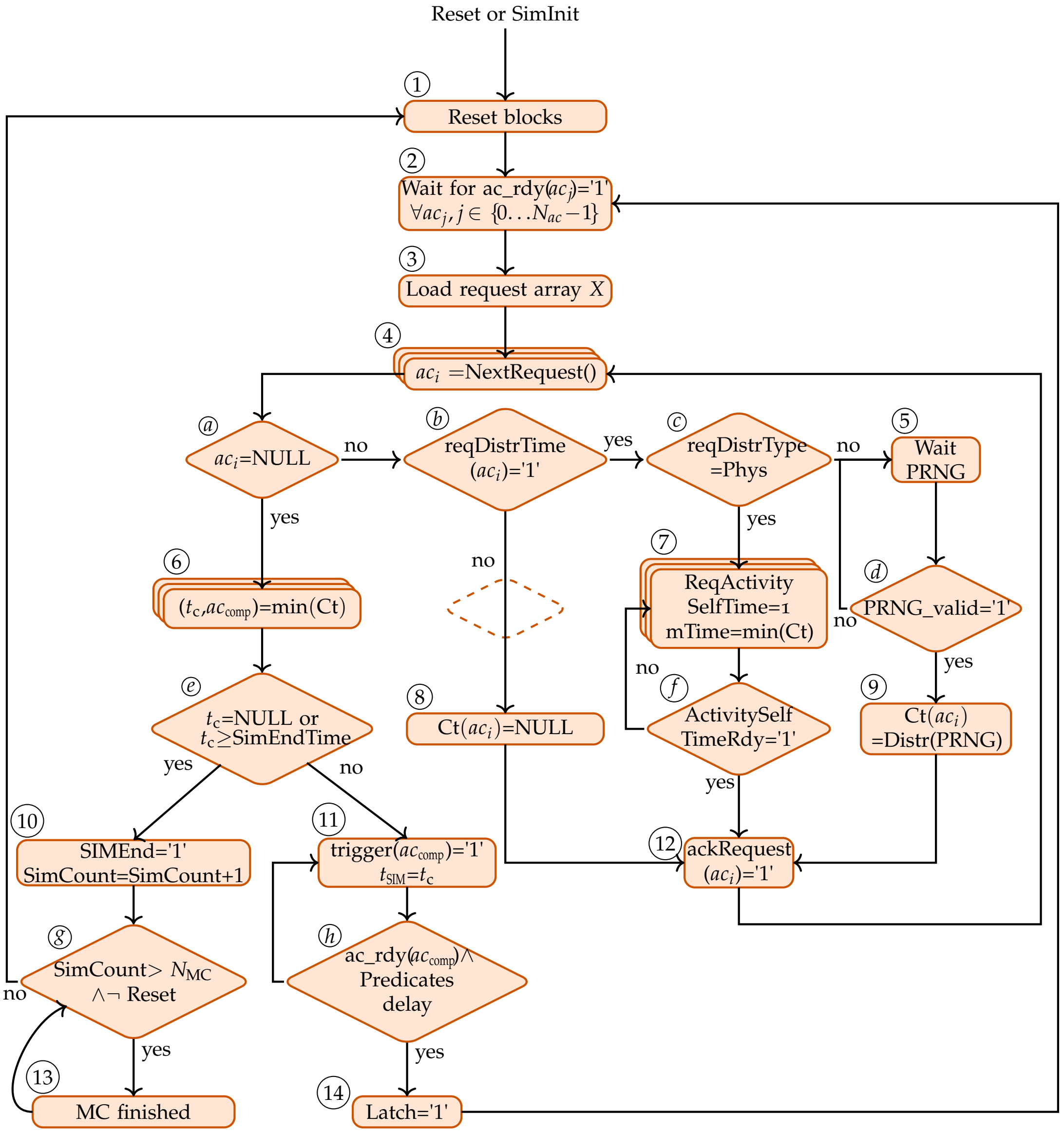


Figure A.5: Flowchart diagram for a parallel SAN simulation.

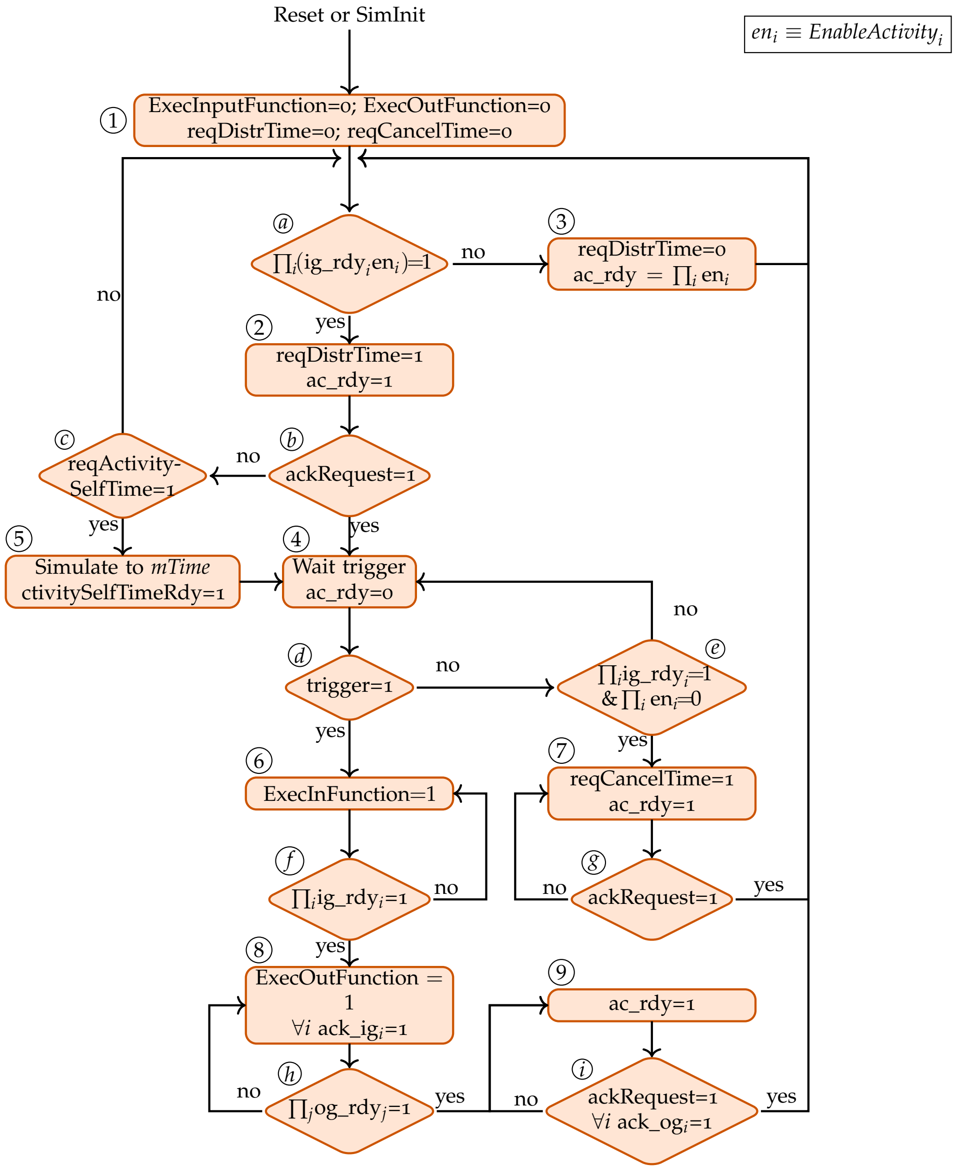


Figure A.6: Flowchart diagram of the activity blocks for a parallel SAN simulation.

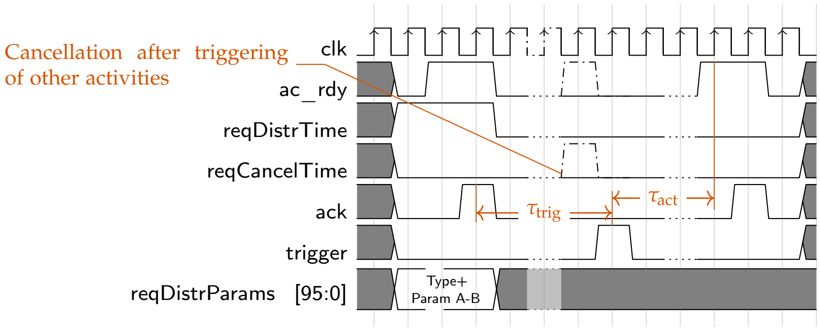


Figure A.7: Signalling between scheduler and activity blocks.

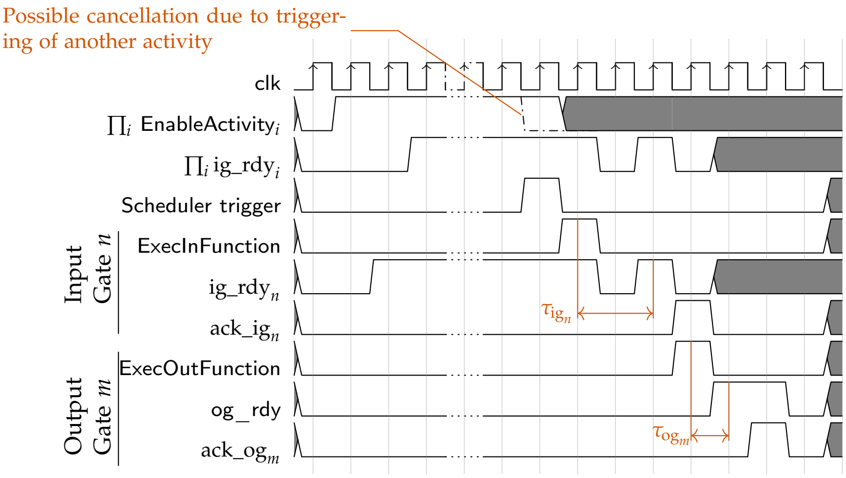


Figure A.8: Signalling between activities and input and output gates.

A.4 RECURRENT METHOD TO EVALUATE THE POINTWISE VARIANCE

Let $X_{k-1} = \{x_1, x_2, \dots, x_{k-1}\}$ be a set of $k-1$ samples of any metric obtained from k trials of a Monte Carlo (MC) simulation. If its variance and expected values are calculated in a previous iteration, the variance of $X_k = X_{k-1} \cup \{x_k\}$ can be calculated as:

$$\mathrm{Var}\left(X_k\right) = \frac{1}{k}\sum_{i}^{k}(x_i - \bar{x}_k)^2 \tag{A.1}$$

where

$$\bar{x}_k = \mathrm{E}\left[X_k\right] \tag{A.2}$$

$$\begin{aligned}
k\mathrm{Var}\left(X_k\right) &= \sum_{i=1}^{k}(x_i - \bar{x}_k)^2 \\
&= \sum_{i=1}^{k}(x_i^2 - 2x_i\bar{x}_k + \bar{x}_k^2) \\
&= \sum_{i=1}^{k-1}(x_i^2) + x_k^2 + \sum_{i=1}^{k}\left(-2x_i\bar{x}_k + \bar{x}_k^2\right) \\
&= \sum_{i=1}^{k-1}\left((x_i - \bar{x}_{k-1}) + \bar{x}_{k-1}\right)^2 + x_k^2 - k\bar{x}_k^2 \\
&= \sum_{i=1}^{k-1}\left(x_i - \bar{x}_{k-1}\right)^2 + \sum_{i=1}^{k-1}\left(2x_i\bar{x}_{k-1} - \bar{x}_{k-1}^2\right) + x_k^2 - k\bar{x}_k^2 \\
&= (k-1)\mathrm{Var}\left(X_{k-1}\right) + \sum_{i=1}^{k-1}\left(2x_i\bar{x}_{k-1} - \bar{x}_{k-1}^2\right) + x_k^2 - k\bar{x}_k^2 \\
&= (k-1)\mathrm{Var}\left(X_{k-1}\right) + (k-1)\bar{x}_{k-1}^2 + x_k^2 - k\bar{x}_k^2
\end{aligned} \tag{A.3}$$

replacing

$$\begin{aligned}
\bar{x}_k &= \frac{\bar{x}_{k-1}(k-1)+x_k}{k} \\
k\bar{x}_k^2 &= \frac{(k-1)^2}{k}\bar{x}_{k-1}^2 + 2\frac{k-1}{k}\bar{x}_{k-1}x_k + \frac{x_k^2}{k} \qquad \text{(A.4)} \\
& \qquad \text{(A.5)}
\end{aligned}$$

results in

$$\begin{aligned}
k\mathrm{Var}\left(X_k\right) &= (k-1)\left[\mathrm{Var}\left(X_{k-1}\right) + \frac{1}{k}\bar{x}_{k-1}^2 + \frac{1}{k}x_k^2 - \frac{2}{k}\bar{x}_{k-1}x_k\right] \\
&= (k-1)\left[\mathrm{Var}\left(X_{k-1}\right) + \frac{\left(x_k-\bar{x}_{k-1}\right)^2}{k}\right]
\end{aligned} \tag{A.6}$$

and

$$\mathrm{Var}\left(X_k\right) = \frac{(k-1)\mathrm{Var}\left(X_{k-1}\right)}{k} + \frac{(k-1)\left(x_k-\bar{x}_{k-1}\right)^2}{k^2} \tag{A.7}$$

A.5 Varn Wormer coefficient simplification

$$\begin{gathered}
p_i\theta_c\left(r_{i,\mathrm{out}}^2 - r_{i,\mathrm{in}}^2\right) + (1-p_i)\theta_{\mathrm{out}}\left(r_{i,\mathrm{out}}^2 - r_{i,\mathrm{in}}^2\right) = \\
\theta_{\mathrm{in}}(r_{i,\mathrm{out}}^2 - r_{i,\mathrm{in}}^2) + \frac{2\left(\theta_{\mathrm{in}}-\theta_{\mathrm{out}}\right)}{\ln\left(\frac{r_{i,\mathrm{in}}}{r_{i,\mathrm{out}}}\right)}\left[-\frac{r_{i,\mathrm{out}}^2 - r_{i,\mathrm{in}}^2}{4} + \frac{r_{i,\mathrm{out}}^2\ln\left(\frac{r_{i,\mathrm{out}}}{r_{i,\mathrm{in}}}\right)}{2}\right]
\end{gathered}$$

$$\begin{gathered}
p_i\left(r_{i,\mathrm{out}}^2 - r_{i,\mathrm{in}}^2\right)\left(\theta_c-\theta_{\mathrm{out}}\right) + \theta_{\mathrm{out}}\left(r_{i,\mathrm{out}}^2 - r_{i,\mathrm{in}}^2\right) = \\
\theta_{\mathrm{in}}(r_{i,\mathrm{out}}^2 - r_{i,\mathrm{in}}^2) + \frac{2\left(\theta_{\mathrm{in}}-\theta_{\mathrm{out}}\right)}{\ln\left(\frac{r_{i,\mathrm{in}}}{r_{i,\mathrm{out}}}\right)}\left[-\frac{r_{i,\mathrm{out}}^2 - r_{i,\mathrm{in}}^2}{4} + \frac{r_{i,\mathrm{out}}^2\ln\left(\frac{r_{i,\mathrm{out}}}{r_{i,\mathrm{in}}}\right)}{2}\right]
\end{gathered}$$

$$p_i\left(\theta_c-\theta_{\mathrm{out}}\right) + \theta_{\mathrm{out}} = \theta_{\mathrm{in}} + \frac{2\left(\theta_{\mathrm{in}}-\theta_{\mathrm{out}}\right)}{\left(r_{i,\mathrm{out}}^2 - r_{i,\mathrm{in}}^2\right)\ln\left(\frac{r_{i,\mathrm{in}}}{r_{i,\mathrm{out}}}\right)}\left[-\frac{r_{i,\mathrm{out}}^2 - r_{i,\mathrm{in}}^2}{4} + \frac{r_{i,\mathrm{out}}^2\ln\left(\frac{r_{i,\mathrm{out}}}{r_{i,\mathrm{in}}}\right)}{2}\right]$$

$$p_i\left(\theta_c-\theta_{\mathrm{out}}\right) + \theta_{\mathrm{out}} = \theta_{\mathrm{in}} + \frac{2\left(\theta_{\mathrm{in}}-\theta_{\mathrm{out}}\right)}{\ln\left(\frac{r_{i,\mathrm{in}}}{r_{i,\mathrm{out}}}\right)}\left[-\frac{1}{4} + \frac{r_{i,\mathrm{out}}^2\ln\left(\frac{r_{i,\mathrm{out}}}{r_{i,\mathrm{in}}}\right)}{2\left(r_{i,\mathrm{out}}^2 - r_{i,\mathrm{in}}^2\right)}\right]$$

$$p_i\left(\theta_c-\theta_{\mathrm{out}}\right) + \theta_{\mathrm{out}} = \theta_{\mathrm{in}} + 2\left(\theta_{\mathrm{in}}-\theta_{\mathrm{out}}\right)\left[-\frac{1}{4\ln\left(\frac{r_{i,\mathrm{in}}}{r_{i,\mathrm{out}}}\right)} - \frac{r_{i,\mathrm{out}}^2}{2\left(r_{i,\mathrm{out}}^2 - r_{i,\mathrm{in}}^2\right)}\right]$$

$$p_i = 1 - 2\left[\frac{1}{4\ln\left(\frac{r_{i,\text{in}}}{r_{i,\text{out}}}\right)} + \frac{r_{i,\text{out}}^2}{2\left(r_{i,\text{out}}^2 - r_{i,\text{in}}^2\right)}\right]$$

$$p_i = 1 - \frac{1}{2\ln\left(\frac{r_{i,\text{in}}}{r_{i,\text{out}}}\right)} - \frac{r_{i,\text{out}}^2}{\left(r_{i,\text{out}}^2 - r_{i,\text{in}}^2\right)}$$

$$p_i = -\frac{1}{2\ln\left(\frac{r_{i,\text{in}}}{r_{i,\text{out}}}\right)} + \frac{\left(r_{i,\text{out}}^2 - r_{i,\text{in}}^2\right) - r_{i,\text{out}}^2}{\left(r_{i,\text{out}}^2 - r_{i,\text{in}}^2\right)}$$

$$p_i = \frac{1}{2\ln\left(\frac{r_{i,\text{out}}}{r_{i,\text{in}}}\right)} - \frac{1}{\left(\frac{r_{i,\text{out}}}{r_{i,\text{in}}}\right)^2 - 1}$$

A.6 Segmented filters with linear regression weight approximates

An alternative approach for the segmentation of the filters that model the mutual-heating and soil effect procresses is using a linear regression for each segment as follows:

$$\begin{aligned}
y_j[n] - y_j[n-1] &= \sum_{i=P_{j-1}-1}^{P_j-1} \Delta W[n-i-1](\beta_{0,j} + \beta_{1,j}(i - P_{j-1} + 1)) \\
&\quad - \sum_{i=P_{j-1}}^{P_j} \Delta W[n-i-1](\beta_{0,j} + \beta_{1,j}(i - P_{j-1})) \\
&= \beta_{0,j} \sum_{i=P_{j-1}-1}^{P_j-1} \Delta W[n-i-1] + \sum_{i=P_{j-1}-1}^{P_j-1} \Delta W[n-i-1]\beta_{1,j}(i - P_{j-1} + 1) \\
&\quad - \beta_{0,j} \sum_{i=P_{j-1}}^{P_j} \Delta W[n-i-1] - \sum_{i=P_{j-1}}^{P_j} \Delta W[n-i-1]\beta_{1,j}(i - P_{j-1}) \\
&= \beta_{0,j}\left(\Delta W[n - P_{j-1} - 2] - \Delta W[n - P_j - 1]\right) \\
&\quad + \beta_{1,j} \sum_{i=P_{j-1}-1}^{P_j-1} \Delta W[n-i-1](i - P_{j-1} + 1) \\
&\quad - \beta_{1,j} \sum_{i=P_{j-1}}^{P_j} \Delta W[n-i-1](i - P_{j-1}) \\
&= \beta_{0,j}\left(\Delta W[n - P_{j-1} - 2] - \Delta W[n - P_j - 1]\right) \\
&\quad + \beta_{1,j}\left(\sum_{i=P_{j-1}-1}^{P_j-1} i\Delta W[n-i-1] - \sum_{i=P_{j-1}}^{P_j} i\Delta W[n-i-1]\right)
\end{aligned}$$

$$
\begin{aligned}
&+\beta_{1,j}\left(P_{j-1}\sum_{i=P_{j-1}}^{P_j}\Delta W[n-i-1]-(P_{j-1}-1)\sum_{i=P_{j-1}-1}^{P_j-1}\Delta W[n-i-1]\right)\\
=\;&\beta_{0,j}\left(\Delta W[n-P_{j-1}-2]-\Delta W[n-P_j-1]\right)\\
&+\beta_{1,j}\left((P_{j-1}-1)\,\Delta W[n-P_{j-1}-2]-P_j\Delta W[n-P_j-1]\right)\\
&+\beta_{1,j}P_{j-1}\left(\sum_{i=P_{j-1}}^{P_j}\Delta W[n-i-1]-\sum_{i=P_{j-1}-1}^{P_j-1}\Delta W[n-i-1]\right)\\
&+\beta_{1,j}\sum_{i=P_{j-1}-1}^{P_j-1}\Delta W[n-i-1]\\
=\;&\left(\beta_{0,j}+\beta_{1,j}\left(P_{j-1}-1\right)\right)\Delta W[n-P_{j-1}-2]-\left(\beta_{0,j}+\beta_{1,j}P_j\right)\Delta W[n-P_j-1]\\
&+\beta_{1,j}P_{j-1}\left(\Delta W[n-P_j-1]-\Delta W[n-P_{j-1}-2]\right)\\
&+\beta_{1,j}\sum_{i=P_{j-1}-1}^{P_j-1}\Delta W[n-i-1]\\
=\;&\left(\beta_{0,j}+\beta_{1,j}\left(P_{j-1}-1\right)-\beta_{1,j}P_{j-1}\right)\Delta W[n-P_{j-1}-2]\\
&-\left(\beta_{0,j}+\beta_{1,j}P_j+\beta_{1,j}P_{j-1}\right)\Delta W[n-P_j-1]\\
&+\beta_{1,j}\sum_{i=P_{j-1}-1}^{P_j-1}\Delta W[n-i-1] \qquad \text{(A.8)}
\end{aligned}
$$

A.7 Exponential integral function and its properties

The exponential integral function $\mathrm{Ei}\left(\cdot\right)$ is a particular case of the function $\mathrm{E}_n\left(\cdot\right)$, defined for complex values as

$$\mathrm{E}_n\left(x\right)=\int_1^{\infty}\frac{e^{-vx}}{v^n}\mathrm{d}v\,,\; n\in\mathbb{N}_0\,,|\arg(x)|<\pi \qquad \text{(A.9)}$$

For $n=1$,

$$\mathrm{E}_1\left(x\right)=\int_1^{\infty}\frac{e^{-vx}}{v}\mathrm{d}v=\int_x^{\infty}\frac{e^{-y}}{v}\mathrm{d}y \qquad \text{(A.10)}$$

and $\mathrm{Ei}\left(\cdot\right)$ is defined for real negative arguments (real positive values of x) as

$$-\mathrm{Ei}\left(-x\right)=\mathrm{E}_1\left(x\right)=\int_x^{\infty}\frac{\mathrm{e}^{-v}}{v}\mathrm{d}v\;,\; x\in\mathbb{R}_{>0} \qquad \text{(A.11)}$$

$\mathrm{E}_1\left(\cdot\right)$ can be expressed as (Abramowitz and Stegun, 1965)

$$\mathrm{E}_1\left(x\right)=-\gamma-\ln x-\sum_{j=1}^{\infty}\frac{(-x)^j}{j\,j!} \qquad \text{(A.12)}$$

and its derivatives can be obtained by applying the property

$$\frac{\mathrm{d}}{\mathrm{d}x}\mathrm{E}_n(x)=-\mathrm{E}_{n-1}(x) \qquad \text{(A.13)}$$

BIBLIOGRAPHY

Abdul Rahman, F. et al. (Mar. 2013). "Application of fault tree analysis for customer reliability assessment of a distribution power system". In: *Reliability Engineering & System Safety* 111, pp. 76–85. ISSN: 09518320. DOI: 10.1016/j.ress.2012.10.011. URL: https://linkinghub.elsevier.com/retrieve/pii/S0951832012002116.

Abramowitz, M. and Stegun, I. A. (1965). *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Vol. 55. Courier Corporation.

AccelerEyes (2019). *ArrayFire*. URL: https://arrayfire.com.

Adelabu, A. J., Haruna, Y., and Aliyu, U. (2018). "Reliability Assessment of Power Equipment in a Distribution Substation Using Fault Tree Analysis". In: *Global Scientific* 6.2, pp. 249–259. URL: http://www.globalscientificjournal.com/researchpaper/Reliability-Assessment-of-Power-Equipment-in-a-Distribution-Substation-Using-Fault-Tree-Analysis.pdf.

Aizpurua, J. I., Catterson, V. M., Papadopoulos, Y., et al. (2017). "Improved Dynamic Dependability Assessment Through Integration With Prognostics". In: *IEEE Transactions on Reliability* 66.3, pp. 893–913. ISSN: 0018-9529. DOI: 10.1109/TR.2017.2693821.

Aizpurua, J. I., Stewart, B. G., et al. (2019). "Towards a data analytics framework for medium voltage power cable lifetime mangement". In:

Aizpurua, J. I. and Catterson, V. M. (2016). "ADEPS : A Methodology for Designing Prognostic Applications". In: *European Conference of the PHM Society 2016*. PHM Society, pp. 1–14.

Aizpurua, J. I., Catterson, V. M., Abdulhadi, I. F., et al. (2018). "A Model-Based Hybrid Approach for Circuit Breaker Prognostics Encompassing Dynamic Reliability and Uncertainty". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48.9, pp. 1637–1648. ISSN: 21682232. DOI: 10.1109/TSMC.2017.2685346.

Aizpurua, J. I., McArthur, S. D. J., et al. (June 2019). "Adaptive Power Transformer Lifetime Predictions Through Machine Learning and Uncertainty Modeling in Nuclear Power Plants". In: *IEEE Transactions on Industrial Electronics* 66.6, pp. 4726–4737. ISSN: 0278-0046. DOI: 10.1109/TIE.2018.2860532. URL: https://ieeexplore.ieee.org/document/8424174/.

Aizpurua, J. and Muxika, E. (2015). "Functionality and dependability assurance in massively networked scenarios". PhD thesis. Mondragon Unibertsitatea. DOI: 10.1201/b15938-269. URL: http://www.crcnetbase.com/doi/abs/10.1201/b15938-269.

Aliee, H. and Zarandi, H. R. (2011). "Fault tree analysis using stochastic logic: A reliable and high speed computing". In: *Reliability and Maintainability Symposium (RAMS), 2011 Proceedings-Annual*. IEEE, pp. 1–6.

Aliee, H. and Zarandi, H. R. (2013). "A fast and accurate fault tree analysis based on stochastic logic implemented on field-programmable gate arrays". In: *IEEE Transactions on Reliability* 62.1, pp. 13–22. ISSN: 00189529. DOI: 10.1109/TR.2012.2221012.

Allen, J. F. (1983). "Maintaining knowledge about temporal intervals". In: *Communications of the ACM* 26.11, pp. 832–843.

Alsharif, M. et al. (2009). "Partial discharge resulting from internal degradation in underground MV cables: Modelling and analysis". In: *2009 44th International Universities Power Engineering Conference (UPEC)*, pp. 1–5.

Alves, G. et al. (May 2019). "A Methodology for Dependability Evaluation of Smart Grids". In: *Energies* 12.9, p. 1817. ISSN: 1996-1073. DOI: 10.3390/en12091817. URL: https://www.mdpi.com/1996-1073/12/9/1817.

Alwan, S. H. et al. (2016). “Factors Affecting Current Ratings for Underground and Air Cables”. In: *World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering* 10.11, pp. 1422–1428.

Amare, T., Helvik, B. E., and Heegaard, P. E. (2018). “A modeling approach for dependability analysis of smart distribution grids”. In: *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE, pp. 1–8. ISBN: 978-1-5386-3458-5. DOI: `10.1109/ICIN.2018.8401634`. URL: `https://ieeexplore.ieee.org/document/8401634/`.

American Furukawa Inc. (2018). *Transmission & Distribution Extra-High Voltage XLPE Cables*. URL: `https://www.furukawaelectric.us/Furukawapdf/EVH%7B%5C_%7DCatalog.pdf` (visited on 09/07/2018).

Ametani, A., Ohno, T., and Nagaoka, N. (June 2015). *Cable System Transients*. Vol. 112. 483. Singapore: John Wiley & Sons, Singapore Pte. Ltd, pp. 211–212. ISBN: 9781118702154. DOI: `10.1002/9781118702154`. URL: `http://doi.wiley.com/10.1002/9781118702154`.

Anders, G. and El-Kady, M. (1992). “Transient ratings of buried power cables. I. Historical perspective and mathematical model”. In: *IEEE Transactions on Power Delivery* 7.4, pp. 1724–1734. ISSN: 08858977. DOI: `10.1109/61.156972`. URL: `http://ieeexplore.ieee.org/document/156972/`.

Anders, G. J. (1997). *Rating of electric power cables : ampacity computations for transmission, distribution, and industrial applications*. Institute of Electrical and Electronics Engineers, p. 428. ISBN: 0780311779.

— (2005). *Rating of Electric Power Cables in Unfavorable Thermal Environment*. Wiley-IEEE Press, p. 326. ISBN: 978-0-471-67909-7. URL: `http://ieeexplore.ieee.org/xpl/bkabstractplus.jsp?bkn=5444115`.

Aras, F., Alekperov, V., et al. (Sept. 2007). “Aging of 154 kV underground power cable insulation under combined thermal and electrical stresses”. In: *IEEE Electrical Insulation Magazine* 23.5, pp. 25–33. ISSN: 0883-7554. DOI: `10.1109/MEI.2007.4318272`. URL: `http://ieeexplore.ieee.org/document/4318272/`.

Aras, F. and Biçen, Y. (2013). “Thermal modelling and analysis of high-voltage insulated power cables under transient loads”. In: *Computer Applications in Engineering Education* 21.3, pp. 516–529. ISSN: 10613773. DOI: `10.1002/cae.20497`.

Arnold, P. and Kment Attila, Pipa Marek, J. F. (2012). “On-site Partial Discharges Measurement of XLPE”. In: 1.4, pp. 107–110.

Arrhenius, S. (1889). “Über die Dissociationswärme und den Einfluss der Temperatur auf den Dissociationsgrad der Elektrolyte”. In: *Zeitschrift für Physikalische Chemie* 4U.1, pp. 96–116. ISSN: 0942-9352. DOI: `10.1515/zpch-1889-0408`.

Arulampalam, M. S. et al. (2002). “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking”. In: *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking* 50.2, pp. 723–737. DOI: `10.1109/9780470544198.ch73`.

Azgomi, M. A. and Movaghar, A. (2003a). “Hierarchical stochastic activity networks”. In: *Proc. of the 17th European Simulation Multiconference-10th Int'l Conf. on Analytical and Stochastic Modelling Techniques and Applications (ESM/ASMTA'03)*, pp. 169–174.

Azgomi, M. A. and Movaghar, A. (2003b). “Coloured Stochastic Activity Networks: Preliminary Definitions and Properties”. In: *Proceedings of the 7th International Workshop on Software and Performance*, pp. 154–169. URL: `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.102.1169`.

Azgomi, M. A., Movaghar, A., et al. (2005). “Hierarchical stochastic activity networks: Formal definitions and behaviour”. In: *International Journal of Simulation: Systems, Science and Technology* 6.1-2, pp. 56–66. ISSN: 1473804X.

Aziz, N. et al. (Oct. 2014). “Prognostic modeling for electrical treeing in solid insulation using pulse sequence analysis”. In: *2014 IEEE Conference on Electrical Insulation and Dielectric Phenomena (CEIDP)*. IEEE, pp. 373–376. ISBN: 978-1-4799-7525-9. DOI: 10.1109/CEIDP.2014.6995906. URL: http://ieeexplore.ieee.org/document/6995906/.

Bahr, N. J. (Dec. 2014). *System Safety Engineering and Risk Assessment*. CRC Press, pp. 1–418. ISBN: 9781466551619. DOI: 10.1201/b17854. URL: https://www.taylorfrancis.com/books/9781466551619.

Bazil Raj, A. A. (2018). *FPGA-Based Embedded System Developer’s Guide*. DOI: 10.1201/9781315156200.

Bejan, A. and Kraus, A. D. (2003). *Heat transfer handbook*. Vol. 1. John Wiley & Sons.

Bicen, Y. (Feb. 2017). “Trend adjusted lifetime monitoring of underground power cable”. In: *Electric Power Systems Research* 143, pp. 189–196. ISSN: 03787796. DOI: 10.1016/j.epsr.2016.10.045. URL: https://linkinghub.elsevier.com/retrieve/pii/S0378779616304424.

Billinton, R. and Li, W. (1994). *Reliability assessment of electric power systems using Monte Carlo methods*. Springer Science & Business Media.

Biswas, S. S., Srivastava, A. K., and Whitehead, D. (June 2015). “A Real-Time Data-Driven Algorithm for Health Diagnosis and Prognosis of a Circuit Breaker Trip Assembly”. In: *IEEE Transactions on Industrial Electronics* 62.6, pp. 3822–3831. ISSN: 0278-0046. DOI: 10.1109/TIE.2014.2362498. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6920077.

Blackman, D. and Vigna, S. (May 2018). “Scrambled Linear Pseudorandom Number Generators”. In: *arXiv preprint arXiv:1805.01407*. arXiv: 1805.01407. URL: http://arxiv.org/abs/1805.01407.

Bobbio, A. et al. (2003). “Parametric fault tree for the dependability analysis of redundant systems and its High-Level Petri Net semantics”. In: *IEEE Transactions on Software Engineering* 29.3, pp. 270–287. ISSN: 00985589. DOI: 10.1109/TSE.2003.1183940.

Bouissou, M. (2002). “Boolean Logic Driven Markov Processes: A Powerful New Formalism for Specifying and Solving Very Large Markov Models”. In: *PSAM 6, June 2002* March, p. 8. DOI: 10.1016/S0951-8320(03)00143-1. URL: http://www-math.univ-mlv.fr/%7B~%7Dbouissou/BDMP%7B%5C_%7DPSAM6.pdf%7B%5C%%7D5CnBouissou%20(2002).pdf.

Buhari, M. and Kopsidas, K. (2014). “Probabilistic assessment of impacts of real-time line ratings on distribution networks”. In: *Probabilistic Methods Applied to Power Systems (PMAPS), 2014 International Conference on*. IEEE, pp. 1–6.

Buhari, M., Levi, V., and Awadallah, S. K. E. (2016). “Modelling of ageing distribution cable for replacement planning”. In: *IEEE Transactions on Power Systems* 31.5, pp. 3996–4004. ISSN: 08858950. DOI: 10.1109/TPWRS.2015.2499269.

Bukowiec, A. and Adamski, M. (2012). “Synthesis of Petri nets into FPGA with operation flexible memories”. In: *Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2012 IEEE 15th International Symposium on*. IEEE, pp. 16–21.

Buller, F. H. (July 1951). “Thermal Transients on Buried Cables”. In: *Transactions of the American Institute of Electrical Engineers* 70.1, pp. 45–55. ISSN: 0096-3860. DOI: 10.1109/T-AIEE.1951.5060370. URL: http://ieeexplore.ieee.org/document/5060370/.

Cadini, F., Agliardi, G. L., and Zio, E. (Jan. 2017a). “A modeling and simulation framework for the reliability/availability assessment of a power transmission grid subject to cascading failures under extreme weather conditions”. In: *Applied Energy* 185, pp. 267–279. ISSN: 03062619. DOI: 10.1016/j.apenergy.2016.10.086. URL: https://linkinghub.elsevier.com/retrieve/pii/S030626191631529X.

Cadini, F., Agliardi, G. L., and Zio, E. (Feb. 2017b). “Estimation of rare event probabilities in power transmission networks subject to cascading failures”. In: *Reliability Engineering & System Safety* 158, pp. 9–20. issn: 09518320. doi: `10.1016/j.ress.2016.09.009`. url: `https://linkinghub.elsevier.com/retrieve/pii/S0951832016305440`.

Calderaro, V. et al. (Oct. 2011). “Failure Identification in Smart Grids Based on Petri Net Modeling”. In: *IEEE Transactions on Industrial Electronics* 58.10, pp. 4613–4623. issn: 0278-0046. doi: `10.1109/TIE.2011.2109335`. url: `http://ieeexplore.ieee.org/document/5704195/`.

Carslaw, H. S. and Jaeger, J. C. (1959). *Conduction of heat in solids: Oxford Science Publications*. Oxford, England.

Catterson, V. M. (Oct. 2014). “Prognostic modeling of transformer aging using Bayesian particle filtering”. In: *2014 IEEE Conference on Electrical Insulation and Dielectric Phenomena (CEIDP)*. IEEE, pp. 413–416. isbn: 978-1-4799-7525-9. doi: `10.1109/CEIDP.2014.6995874`. url: `http://ieeexplore.ieee.org/document/6995874/`.

Catterson, V. M., Melone, J., and Garcia, M. S. (Jan. 2016). “Prognostics of transformer paper insulation using statistical particle filtering of on-line data”. In: *IEEE Electrical Insulation Magazine* 32.1, pp. 28–33. issn: 0883-7554. doi: `10.1109/MEI.2016.7361101`. url: `http://ieeexplore.ieee.org/document/7361101/`.

Čepin, M. and Mavko, B. (Jan. 2002). “A dynamic fault tree”. In: *Reliability Engineering & System Safety* 75.1, pp. 83–91. issn: 09518320. doi: `10.1016/S0951-8320(01)00121-1`. url: `https://linkinghub.elsevier.com/retrieve/pii/S0951832001001211`.

Cerdà-Alabern, L. (2014). “Closed Form Transient Solution of Continuous Time Markov Chains Through Uniformization”. In: *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools*. ICST (Institute for Computer Sciences, Social-Informatics and˜… ICST, pp. 263–272. isbn: 978-1-936968-48-0. doi: `10.4108/icst.valuetools.2013.254376`. url: `http://eudl.eu/doi/10.4108/icst.valuetools.2013.254376`.

Chakravorti, S., Dey, D., and Chatterjee, B. (2013). *Recent Trends in the Condition Monitoring of Transformers*. Vol. 67. Power Systems. London: Springer London. isbn: 978-1-4471-5549-2. doi: `10.1007/978-1-4471-5550-8`. url: `http://link.springer.com/10.1007/978-1-4471-5550-8`.

Chang, N., Kwon, W. H., and Park, J. (1996). “FPGA-based implementation of synchronous Petri nets”. In: *Industrial Electronics, Control, and Instrumentation, 1996., Proceedings of the 1996 IEEE IECON 22nd International Conference on*. Vol. 1. IEEE, pp. 469–474.

Chen, B. et al. (2017). “Recognition of high-voltage cable partial discharge signal based on adaptive fuzzy c-means clustering”. In: *International Journal of Pattern Recognition and Artificial Intelligence* 31.6, p. 1759009. issn: 02180014. doi: `10.1142/S0218001417590091`.

Chen, K.-L. and Chen, N. (Jan. 2011). “A New Method for Power Current Measurement Using a Coreless Hall Effect Current Transformer”. In: *IEEE Transactions on Instrumentation and Measurement* 60.1, pp. 158–169. issn: 0018-9456. doi: `10.1109/TIM.2010.2049234`. url: `http://ieeexplore.ieee.org/document/5471219/`.

Cheng, S. and Pecht, M. (2007). “Multivariate State Estimation Technique for remaining useful life prediction of electronic products”. In: *AAAI Fall Symposium - Technical Report* 1, pp. 26–32.

Chiaradonna, S., Lollini, P., and Di Giandomenico, F. (2007). “On a modeling framework for the analysis of interdependencies in electric power systems”. In: *Dependable Systems and Networks, 2007. DSN'07. 37th Annual IEEE/IFIP International Conference on*. IEEE, pp. 185–195.

Choonhapran, P. and Balzer, G. (June 2007). “Availability of HV Circuit-Breakers: The Application of Markov Model”. In: *2007 IEEE Power Engineering Society General Meet-*

ing. IEEE, pp. 1–6. ISBN: 1-4244-1296-X. DOI: `10.1109/PES.2007.385673`. URL: `http://ieeexplore.ieee.org/document/4275282/`.

Chraibi, H. (2013). “Dynamic reliability modeling and assessment with PyCATSHOO: Application to a test case”. In: *PSAM congress*.

Coble, J. B. et al. (July 2012). *Prognostics and Health Management in Nuclear Power Plants: A Review of Technologies and Applications*. Tech. rep. July. Richland, WA (United States): Pacific Northwest National Laboratory (PNNL), pp. 1–22. DOI: `10.2172/1047416`. URL: `http://www.pnnl.gov/main/publications/external/technical%7B%5C_%7Dreports/PNNL-21515.pdf`.

Codetta-Raiteri, D. (May 2011). “Integrating several formalisms in order to increase Fault Trees’ modeling power”. In: *Reliability Engineering & System Safety* 96.5, pp. 534–544. ISSN: 09518320. DOI: `10.1016/j.ress.2010.12.027`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0951832010002838`.

Cong, J. et al. (2011). “High-level synthesis for FPGAs: From prototyping to deployment”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30.4, pp. 473–491.

Crine, J.-P. P. (2005). “On the interpretation of some electrical aging and relaxation phenomena in solid dielectrics”. In: *IEEE Transactions on Dielectrics and Electrical Insulation* 12.6, pp. 1089–1107. ISSN: 10709878. DOI: `10.1109/TDEI.2005.1561789`.

Crine, J.-P. (1998). “Electrical, chemical and mechanical processes in water treeing”. In: *IEEE Transactions on Dielectrics and Electrical Insulation* 5.5, pp. 681–694. ISSN: 10709878. DOI: `10.1109/94.729690`. URL: `http://ieeexplore.ieee.org/document/729690/`.

Da Silva, E. et al. (1998). “Diagnosis of XLPE insulated cables aged under conditions of multiple stresses: Thermoelectric and humidity”. In: *Conference Record of IEEE International Symposium on Electrical Insulation* 1, pp. 117–121. ISSN: 01642006.

Dakin, T. W. (1948). “Electrical insulation deterioration treated as a chemical rate phenomenon”. In: *Transactions of the American Institute of Electrical Engineers* 67.1, pp. 113–122.

Danger, J.-L. J.-L. et al. (2000). “Efficient FPGA implementation of Gaussian noise generator for communication channel emulation”. In: *Electronics, Circuits and Systems, 2000. ICECS 2000. The 7th IEEE International Conference on*. Vol. 1. IEEE, pp. 366–369. URL: `https://hal.archives-ouvertes.fr/hal-00347213`.

Diaz-Aguilo, M. and Leon, F. de (2015). “Introducing Mutual Heating Effects in the Ladder-Type Soil Model for the Dynamic Thermal Rating of Underground Cables”. In: *IEEE Transactions on Power Delivery* 30.4, pp. 1958–1964. ISSN: 0885-8977. DOI: `10.1109/TPWRD.2015.2390072`. URL: `http://ieeexplore.ieee.org/document/7008559/`.

Diaz-Aguiló, M. and León, F. de (2015). “Adaptive soil model for real-time thermal rating of underground power cables”. In: *IET Science, Measurement & Technology* 9.6, pp. 654–660. ISSN: 1751-8822. DOI: `10.1049/iet-smt.2014.0269`. URL: `https://digital-library.theiet.org/content/journals/10.1049/iet-smt.2014.0269`.

Dinavahi, V. and Chen, Y. (May 2013). “Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems”. In: *IET Generation, Transmission & Distribution* 7.5, pp. 451–463. ISSN: 1751-8687. DOI: `10.1049/iet-gtd.2012.0374`. URL: `https://digital-library.theiet.org/content/journals/10.1049/iet-gtd.2012.0374`.

Djeziri, M., Benmoussa, S., and Benbouzid, M. (Nov. 2019). “Data-driven approach augmented in simulation for robust fault prognosis”. In: *Engineering Applications of Artificial Intelligence* 86, pp. 154–164. ISSN: 09521976. DOI: `10.1016/j.engappai.2019.09.002`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0952197619302180`.

Doucet, A., Freitas, N. de, and Gordon, N. J. (2001). *Sequential Monte Carlo Methods in Practice. Series Statistics For Engineering and Information Science*.

Douglass, D. and Edris, A.-A. (July 1996). "Real-time monitoring and dynamic thermal rating of power transmission circuits". In: *IEEE Transactions on Power Delivery* 11.3, pp. 1407–1418. issn: 08858977. doi: `10.1109/61.517499`. url: `http://ieeexplore.ieee.org/document/517499/`.

Dueñas-Osorio, L. and Vemuru, S. M. (Mar. 2009). "Cascading failures in complex infrastructure systems". In: *Structural Safety* 31.2, pp. 157–167. issn: 01674730. doi: `10.1016/j.strusafe.2008.06.007`. url: `https://linkinghub.elsevier.com/retrieve/pii/S016747300800057X`.

Echavarren, F. M., Rouco, L., and González, A. (2011). "Dynamic thermal modeling of isolated cables". In: *17th Power System Computation Conference (PSCC'11), Stockholm, Sweden*.

Ejlali, A. and Miremadi, S. G. (2004). "FPGA-based Monte Carlo simulation for fault tree analysis". In: *Microelectronics Reliability* 44.6, pp. 1017–1028. issn: 00262714. doi: `10.1016/j.microrel.2004.01.016`.

Ekwaro-Osire, S., Gonçalves, A. C., and Alemayehu, F. M. (2017). *Probabilistic Prognostics and Health Management of Energy Systems*. Springer.

Engel, S. J. et al. (2000). "Prognostics, the real issues involved with predicting life remaining". In: *IEEE Aerospace Conference Proceedings* 6, pp. 457–470. issn: 1095323X. doi: `10.1109/AERO.2000.877920`.

Feizifar, B. and Usta, O. (June 2019). "A Novel Arcing Power-Based Algorithm for Condition Monitoring of Electrical Wear of Circuit Breaker Contacts". In: *IEEE Transactions on Power Delivery* 34.3, pp. 1060–1068. issn: 0885-8977. doi: `10.1109/TPWRD.2018.2882013`. url: `https://ieeexplore.ieee.org/document/8540028/`.

Fibich, C. et al. (2018). "Preliminary Evaluation of High-level Synthesis Tools-Xilinx Vivado and PandA Bambu". In: *2018 IEEE 13th International Symposium on Industrial Embedded Systems, SIES 2018 - Proceedings*. isbn: 9781538641552. doi: `10.1109/SIES.2018.8442100`.

Galtier, T. (2019). "Accelerated Monte-Carlo methods for Piecewise Deterministic Markov Processes for a faster reliability assessment of power generation systems within the PyCATSHOO toolbox". PhD thesis. Université de Paris, p. 170.

Garro, U. et al. (Apr. 2019a). "FPGA-Based Degradation and Reliability Monitor for Underground Cables". In: *Sensors* 19.9, p. 1995. issn: 1424-8220. doi: `10.3390/s19091995`. url: `https://www.mdpi.com/1424-8220/19/9/1995`.

Garro, U. et al. (July 2019b). "FPGA-Based Stochastic Activity Networks for On-Line Reliability Monitoring". In: *IEEE Transactions on Industrial Electronics*, pp. 1–1. issn: 0278-0046. doi: `10.1109/TIE.2019.2928244`. url: `https://ieeexplore.ieee.org/document/8765588/`.

Gautham, S. (2010). "An Efficient Implementation of an Exponential Random Number Generator in a Field Programmable Gate Array (FPGA)". In:

Goldenberg, H. (1967). "Thermal transients in linear systems with heat generation linearly temperature-dependent. Application to buried cables". In: *Proceedings of the Institution of Electrical Engineers* 114.3, p. 375. issn: 00203270. doi: `10.1049/piee.1967.0081`.

Goldfeld, A. and Dubi, A. (1987). "Monte carlo methods in reliability engineering". In: *Quality and Reliability Engineering International* 3.2, pp. 83–91. issn: 10991638. doi: `10.1002/qre.4680030205`.

Gönen, T. (2014). *Electric Power Distribution Engineering, Third Edition*. CRC Press. isbn: 9781482207057. doi: `10.1201/b16455`.

Guo, H. et al. (Oct. 2018). "A Particle Filtering Based Approach for Transformer Winding Degradation Prognostics". In: *2018 Prognostics and System Health Management Confer-*

ence (PHM-Chongqing). IEEE, pp. 697–703. isbn: 978-1-5386-5380-7. doi: 10.1109/PHM-Chongqing.2018.00125. url: https://ieeexplore.ieee.org/document/8603433/.

Gutierrez, R., Torres, V., and Valls, J. (2012). "Hardware architecture of a gaussian noise generator based on the inversion method". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 59.8, pp. 501–505. issn: 15497747. doi: 10.1109/TCSII.2012.2204119.

Hahn, D. W. and Özişik, M. N. (2012). *Heat Conduction*. Hoboken, NJ, USA: John Wiley & Sons, Inc. isbn: 9781118411285. doi: 10.1002/9781118411285. url: http://doi.wiley.com/10.1002/9781118411285.

Han, Y. J., Lee, H. M., and Shin, Y. J. (2017). "Thermal aging estimation with load cycle and thermal transients for XLPE-insulated underground cable". In: *2017 IEEE Conference on Electrical Insulation and Dielectric Phenomenon (CEIDP)*, pp. 205–208. doi: 10.1109/CEIDP.2017.8257566.

Hao, Y. et al. (Feb. 2018). "Icing Condition Assessment of In-Service Glass Insulators Based on Graphical Shed Spacing and Graphical Shed Overhang". In: *Energies* 11.2, p. 318. issn: 1996-1073. doi: 10.3390/en11020318. url: http://www.mdpi.com/1996-1073/11/2/318.

Hassane CHRAIBI (2019). *PyCATSHOO*. url: http://pycatshoo.org/ (visited on 11/01/2019).

Hauschild, W. and Mosch, W. (1992). *Statistical techniques for high-voltage engineering*. 13. IET.

Hellesø, S. M. et al. (2011). "Calculation of water ingress in a HV subsea XLPE cable with a layered water barrier sheath system". In: *Journal of Applied Polymer Science* 121.4, pp. 2127–2133.

Horton, G. et al. (1998). "Fluid stochastic petri nets: Theory, applications, and solution techniques". In: *European Journal of Operational Research* 105.1, pp. 184–201. issn: 03772217. doi: 10.1016/S0377-2217(97)00028-3.

Hosseini, M. et al. (Oct. 2018). "Current-Based Trip Coil Analysis of Circuit Breakers for Fault Diagnosis". In: *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE, pp. 1–6. isbn: 978-1-5386-4505-5. doi: 10.1109/ISGTEurope.2018.8571478. url: https://ieeexplore.ieee.org/document/8571478/.

Hyvönen, P. et al. (2008). "Prediction of insulation degradation of distribution power cables based on chemical analysis and electrical measurements". PhD thesis. isbn: 9789512294022. url: https://aaltodoc.aalto.fi/handle/123456789/4498.

IEC (2006). *Electric Cables–Calculation of Current Rating–Part 1-1: Current rating equations (100% load factor) and calculation of losses - General*. Standard IEC60287-1-1:2006/AMD1:2014. International Electrotechnical Commission, p. 65.

— (2008). *Calculation of the cyclic and emergency current rating of cables. Part 2: Cyclic rating of cables greater than 18/30 (36) kV and emergency ratings for cables of all voltages.* Standard IEC60853-2:1989/A1:2008. International Electrotechnical Commission, p. 104.

IEEE (2012). *1666-2011 - IEEE standard for standard SystemC language reference manual*. doi: 10.1109/IEEESTD.2012.6134619. url: https://ieeexplore.ieee.org/document/6134619.

ISO (2012). *ISO 13372:2012 - Condition monitoring and diagnostics of machines — Vocabulary*. url: https://www.iso.org/standard/52256.html.

Jee Keen Raymond, W., Illias, H. A., and Abu Bakar, A. H. (Jan. 2017). "Classification of Partial Discharge Measured under Different Levels of Noise Contamination". In: *PLOS ONE* 12.1. Ed. by Gao, Z.-K., e0170111. issn: 1932-6203. doi: 10.1371/journal.pone.0170111. url: https://dx.plos.org/10.1371/journal.pone.0170111.

Jiang, Z. and Willson, A. N. (1995). "Design and implementation of efficient pipelined IIR digital filters". In: *IEEE transactions on signal processing* 43.3, pp. 579–590.

Kabir, S. (July 2017). “An overview of fault tree analysis and its application in model based dependability analysis”. In: *Expert Systems with Applications* 77, pp. 114–135. ISSN: 09574174. DOI: `10.1016/j.eswa.2017.01.058`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0957417417300714`.

Kabir, S., Walker, M., and Papadopoulos, Y. (2014). “Reliability analysis of dynamic systems by translating temporal fault trees into Bayesian networks”. In: *Model-Based Safety and Assessment*. Springer, pp. 96–109.

Kennelly, A. E. (1893). “On the carrying capacity of electric cables submerged buried or suspended in air”. In: *Minutes, Ninth Annual Meeting*.

Khan, A. A. et al. (2012). “A review of condition monitoring of underground power cables”. In: *Proceedings of 2012 IEEE International Conference on Condition Monitoring and Diagnosis, CMD 2012* September, pp. 909–912. DOI: `10.1109/CMD.2012.6416300`.

Kim, N.-H., An, D., and Choi, J.-H. (2017a). *Prognostics and Health Management of Engineering Systems*. Cham: Springer International Publishing, pp. 1–345. ISBN: 978-3-319-44740-7. DOI: `10.1007/978-3-319-44742-1`. URL: `http://link.springer.com/10.1007/978-3-319-44742-1`.

— (2017b). “Prognostics and health management of engineering systems”. In: *Switzerland: Springer International Publishing*.

Kopsidas, K. and Liu, S. (Mar. 2018). “Power Network Reliability Framework for Integrating Cable Design and Ageing”. In: *IEEE Transactions on Power Systems* 33.2, pp. 1521–1532. ISSN: 0885-8950. DOI: `10.1109/TPWRS.2017.2720196`. URL: `https://ieeexplore.ieee.org/document/8003450/`.

Kramer, M. A. (1992). “Autoassociative neural networks”. In: *Computers & chemical engineering* 16.4, pp. 313–328.

Kroese, D. P., Taimre, T., and Botev, Z. I. (2011). *Handbook of Monte Carlo Methods*. ISBN: 9781118014967. DOI: `10.1002/9781118014967`.

Kumamoto, H. and Henley, E. J. (1996). “Probabilistic risk assessment and management for engineers and scientists”. In: *IEE Press, New York, USA*.

Lee, D. U. et al. (2005). “A hardware Gaussian noise generator using the Wallace method”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 13.8, pp. 911–919. ISSN: 10638210. DOI: `10.1109/TVLSI.2005.853615`.

Leone, G., Cristaldi, L., and Turrin, S. (Oct. 2017). “A data-driven prognostic approach based on statistical similarity: An application to industrial circuit breakers”. In: *Measurement* 108, pp. 163–170. ISSN: 02632241. DOI: `10.1016/j.measurement.2017.02.017`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0263224117301100`.

Leroux, H., Andreu, D., and Godary-Dejean, K. (2015). “Handling exceptions in Petri net-based digital architecture: from formalism to implementation on FPGAs”. In: *IEEE Transactions on Industrial Informatics* 11.4, pp. 897–906.

Li, G. et al. (2017). “Large Power Transformer Fault Diagnosis and Prognostic Based on DBNC and D-S Evidence Theory”. In: *Energy and Power Engineering* 09.04, pp. 232–239. ISSN: 1949-243X. DOI: `10.4236/epe.2017.94B028`. URL: `http://www.scirp.org/journal/PaperDownload.aspx?DOI=10.4236/epe.2017.94B028`.

Lindquist, T., Bertling, L., and Eriksson, R. (2008). “Circuit breaker failure data and reliability modelling”. In: *IET Generation, Transmission & Distribution* 2.6, p. 813. ISSN: 17518687. DOI: `10.1049/iet-gtd:20080127`. URL: `https://digital-library.theiet.org/content/journals/10.1049/iet-gtd%7B%5C_%7D20080127`.

Liu, J. et al. (2010). “An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries”. In: *Annual Conference of the Prognostics and Health Management Society, PHM 2010*, pp. 1–8.

Liu, S., Mingas, G., and Bouganis, C.-S. (2017). "An Unbiased MCMC FPGA-Based Accelerator in the Land of Custom Precision Arithmetic". In: *IEEE Transactions on Computers* 66.5, pp. 745–758.

Makhkamova, I. (2011). "Numerical Investigations of the Thermal State of Overhead Lines and Underground Cables in Distribution Networks". In:

Mamedov, S. et al. (2004). "Effects of mechanical and thermal stresses on electric degradation of polyolefins and related materials". In: *Macromolecular Symposia*. Vol. 212. 1. Wiley Online Library, pp. 293–298. DOI: 10.1002/masy.200450833.

Manno, G. et al. (2014). "Conception of Repairable Dynamic Fault Trees and resolution by the use of RAATSS, a Matlab® toolbox based on the ATS formalism". In: *Reliability Engineering & System Safety* 121, pp. 250–262. ISSN: 09518320. DOI: 10.1016/j.ress.2013.09.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S0951832013002597.

Manno, G. A. (2012). "Reliability modelling of complex systems: an adaptive transition system approach to match accuracy and efficiency". PhD thesis.

Manolakis, J. G. P. D. G. (2013). *Digital Signal Processing*. Internatio. Pearson Education. ISBN: 1292025735,9781292025735.

Mariut, L. et al. (2012). "Thermal analysis of underground power cables-A monitoring procedure". In: *Applied and Theoretical Electricity (ICATE), 2012 International Conference on*. IEEE, pp. 1–6.

Marsaglia, G. (2003). "Xorshift RNGs". In: *Journal of Statistical Software* 8.14, pp. 1–6. ISSN: 1548-7660. DOI: 10.18637/jss.v008.i14. URL: http://www.jstatsoft.org/v08/i14/.

Marsaglia, G., Tsang, W. W., et al. (2000). "The ziggurat method for generating random variables". In: *Journal of Statistical Software* 5.8, pp. 1–7. ISSN: 15487660.

Marzinotto, M. and Mazzanti, G. (2014). "Merging the electro-thermal life model for power cables with the statistical volume enlargement law". In: *2014 IEEE Conference on Electrical Insulation and Dielectric Phenomena, CEIDP 2014* 3, pp. 502–505. DOI: 10.1109/CEIDP.2014.6995888.

Matsumoto, M. and Nishimura, T. (Jan. 1998). "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator". In: *ACM Transactions on Modeling and Computer Simulation* 8.1, pp. 3–30. ISSN: 10493301. DOI: 10.1145/272991.272995. URL: http://portal.acm.org/citation.cfm?doid=272991.272995.

Maza, S. (2015). "Diagnosis Modelling for Dependability Assessment of Fault-Tolerant Systems Based on Stochastic Activity Networks". In: *Quality and Reliability Engineering International* 31.6, pp. 963–976. ISSN: 10991638. DOI: 10.1002/qre.1652.

Mazzanti, G. (2009). "The combination of electro-thermal stress, load cycling and thermal transients and its effects on the life of high voltage ac cables". In: *IEEE Transactions on Dielectrics and Electrical Insulation* 16.4, pp. 1168–1179. ISSN: 10709878. DOI: 10.1109/TDEI.2009.5211872.

Mazzanti, G. and Montanari, G. C. (1997). "A comparison between XLPE and EPR as insulating materials for HV cables". In: *IEEE Power Engineering Review* 17.1, p. 36. ISSN: 02721724. DOI: 10.1109/MPER.1997.560656. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=568220.

Mazzanti, G. (2007). "Analysis of the combined effects of load cycling, thermal transients, and electro-thermal stress on life expectancy of high-voltage AC cables". In: *IEEE Transactions on Power Delivery* 22.4, pp. 2000–2009. ISSN: 08858977. DOI: 10.1109/TPWRD.2007.905547.

Mazzanti, G. and Marzinotto, M. (2013). *Extruded cables for high-voltage direct-current transmission: advances in research and development*. Wiley-IEEE Press, p. 384. ISBN: 978-1-118-09666-6.

McCollum, J. M. et al. (2003). “Hardware acceleration of pseudo-random number generation for simulation applications”. In: *Proceedings of the Annual Southeastern Symposium on System Theory* 2003-Janua.X, pp. 299–303. doi: 10.1109/SSST.2003.1194578.

Meeker, W. Q. and Escobar, L. A. (1998). *Statistical methods for reliability data*. Wiley, p. 680. isbn: 9781118625972. url: https://www.wiley.com/en-es/Statistical+Methods+for+Reliability+Data-p-9781118625972.

Megger, Megger, C. B., and Megger (2012). “Circuit Breaker Testing Guide”. In: *Technical Guide*, p. 52. url: www.megger.com.

Mehl, R. and Meckler, P. (Sept. 2005). “Modular Conventional Protection and its Enhancemen through Electronic Circuit Breaker Systems”. In: *INTELEC 05 - Twenty-Seventh International Telecommunications Conference*. IEEE, pp. 65–70. isbn: 978-3-8007-2905-0. doi: 10.1109/INTLEC.2005.335195. url: http://ieeexplore.ieee.org/document/4134308/.

Mentor Graphics (2019). *Catapult HLS*. url: https://www.mentor.com/hls-lp/catapult-high-level-synthesis/ (visited on 06/11/2019).

Meyer-Baese, U. (2014). *Digital Signal Processing with Field Programmable Gate Arrays*. Springer.

Millar, R. J. (2006). “a Comprehensive Approach To Real Time Power Cable Temperature Prediction and Rating in Thermally Unstable Environments”. PhD thesis, pp. 1–157. isbn: 9789512284153.

Milne, I., Ritchie, R. O., and Karihaloo, B. L. (2003). *Comprehensive structural integrity: Cyclic loading and fatigue*. Vol. 4. Elsevier.

Montanari, G. C., Mazzanti, G., and Simoni, L. (2002). “Progress in electrothermal life modeling of electrical insulation during the last decades”. In: *IEEE Transactions on Dielectrics and Electrical Insulation* 9.5, pp. 730–745. issn: 10709878. doi: 10.1109/TDEI.2002.1038660.

Movaghar, A. (1984). “Performability modeling with stochastic activity networks”. In: pp. 215–224.

— (2001). “Stochastic activity networks: A new definition and some properties”. In: *Scientia Iranica* 8.4, pp. 303–311. issn: 10263098.

Neher, J. H. (Feb. 1964). “The Transient Temperature Rise of Buried Cable Systems”. In: *IEEE Transactions on Power Apparatus and Systems* 83.2, pp. 102–114. issn: 0018-9510. doi: 10.1109/TPAS.1964.4765976. url: http://ieeexplore.ieee.org/document/4765976/.

Nelson, B. L. (2013). *Foundations and Methods of Stochastic Simulation*. Vol. 187. International Series in Operations Research & Management Science. Boston, MA: Springer US, p. 67. isbn: 978-1-4614-6159-3. doi: 10.1007/978-1-4614-6160-9. url: http://link.springer.com/10.1007/978-1-4614-6160-9.

Nikulin, M. et al., eds. (2010). *Advances in Degradation Modeling*. Boston, MA: Birkhäuser Boston. isbn: 978-0-8176-4923-4. doi: 10.1007/978-0-8176-4924-1. url: http://link.springer.com/10.1007/978-0-8176-4924-1.

Nurvitadhi, E. et al. (2017). “Can FPGAs beat GPUs in accelerating next-generation deep neural networks?” In: *FPGA 2017 - Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 5–14. doi: 10.1145/3020078.3021740.

NVIDIA Corporation (2019). *CUDA Zone*. url: https://developer.nvidia.com/cuda-zone.

Ogata, K. (1994). *Discrete-Time Control Systems*. Pearson Education. isbn: 0133286428.

Omondi, A. R. (1999). *The Microarchitecture of Pipelined and Superscalar Computers*. Boston, MA: Springer US. isbn: 978-1-4419-5081-9. doi: 10.1007/978-1-4757-2989-4. url: http://link.springer.com/10.1007/978-1-4757-2989-4.

Oppenheim, A. V. et al. (1996). *Signals and systems*. 2nd ed. Prentice Hall. isbn: 978-0136511755. url: https://dl.acm.org/citation.cfm?id=248702.

Orchard, M. E. and Vachtsevanos, G. J. (2009). “A particle-filtering approach for on-line fault diagnosis and failure prognosis”. In: *Transactions of the Institute of Measurement and Control* 31.3-4, pp. 221–246. issn: 01423312. doi: 10.1177/0142331208092026.

Orchard, M., Tobar, F., and Vachtsevanos, G. (2009). “Outer Feedback Correction Loops in Particle Filtering-Based Prognostic Algorithms: Statistical Performance Comparison”. In: *Studies in Informatics and Control* 18.4, pp. 295–304. issn: 1220-1766.

Ortega-Zamorano, F. et al. (Sept. 2016). “FPGA Hardware Acceleration of Monte Carlo Simulations for the Ising Model”. In: *IEEE Transactions on Parallel and Distributed Systems* 27.9, pp. 2618–2627. issn: 1045-9219. doi: 10.1109/TPDS.2015.2505725. url: http://ieeexplore.ieee.org/document/7347434/.

Palshikar, G. K. (2002). “Temporal fault trees”. In: *Information and Software Technology* 44.3, pp. 137–150. issn: 09505849. doi: 10.1016/S0950-5849(01)00223-3.

Pan, W., Wang, Z., and Sun, B. (2015). “Comparison of stochastic response surface method and Monte Carlo method for uncertainty analysis of electronics prognostics”. In: *Prognostics and Health Management (PHM), 2015 IEEE Conference on*. IEEE, pp. 1–7.

PandA Bambu (2019). url: https://panda.dei.polimi.it/?page%7B%5C_%7Did=31 (visited on 06/11/2019).

Papakonstantinou, A. et al. (July 2009). “FCUDA: Enabling efficient compilation of CUDA kernels onto FPGAs”. In: *2009 IEEE 7th Symposium on Application Specific Processors*. IEEE, pp. 35–42. isbn: 978-1-4244-4939-2. doi: 10.1109/SASP.2009.5226333. url: http://ieeexplore.ieee.org/document/5226333/.

Pecht, M. (2008). *Prognostics and health management of electronics*. Wiley Online Library.

Petercem Sensors (2018). *Current sensors, voltage sensors and voltage detectors*. url: http://www.petercem.com/wp-content/uploads/2018/02/catalog-Sensors-A1%7B%5C_%7DGB-BD.pdf.

Piriou, P.-Y. Y., Faure, J.-M. M., and Lesage, J.-J. J. (2016). “From safety analysis of reconfigurable systems to design of fault-tolerant control strategies”. In: *Conference on Control and Fault-Tolerant Systems, SysTol* 2016-Novem, pp. 624–629. issn: 21621209. doi: 10.1109/SYSTOL.2016.7739818.

Qingyue, Y. et al. (Apr. 2019). “Properties of Water Tree Growing in XLPE and composites”. In: *2019 2nd International Conference on Electrical Materials and Power Equipment (ICEMPE)*. IEEE, pp. 409–412. isbn: 978-1-5386-8434-4. doi: 10.1109/ICEMPE.2019.8727376. url: https://ieeexplore.ieee.org/document/8727376/.

Raiteri, D. et al. (Jan. 2004). “Repairable fault tree for the automatic evaluation of repair policies”. In: *International Conference on Dependable Systems and Networks, 2004*. IEEE. IEEE, pp. 659–668. isbn: 0-7695-2052-9. doi: 10.1109/DSN.2004.1311936. url: http://ieeexplore.ieee.org/document/1311936/.

Red Eléctrica de España (2019). *Red Eléctrica de España-ESIOS*. url: https://www.esios.ree.es/en/analysis/1293 (visited on 11/23/2017).

Rerak, M. and Ocłoń, P. (Feb. 2017). “The effect of soil and cable backfill thermal conductivity on the temperature distribution in underground cable system”. In: *E3S Web of Conferences* 13, p. 02004. issn: 2267-1242. doi: 10.1051/e3sconf/20171302004. url: http://www.e3s-conferences.org/10.1051/e3sconf/20171302004.

Risos, A. and Gouws, G. (May 2019). “In-situ aging monitoring of transformer oil via the relative permittivity and DC conductivity using novel interdigitated dielectrometry sensors (IDS)”. In: *Sensors and Actuators B: Chemical* 287, pp. 602–610. issn: 09254005. doi: 10.1016/j.snb.2018.12.037. url: https://linkinghub.elsevier.com/retrieve/pii/S0925400518321592.

Roberts, D., Taylor, P., and Michiorri, A. (2008). “Dynamic thermal rating for increasing network capacity and delaying network reinforcements”. In: *CIRED Seminar 2008: Smart-*

Grids for Distribution. IEE, pp. 61–61. ISBN: 978 0 86341 935 5. DOI: 10.1049/ic:20080466. URL: https://digital-library.theiet.org/content/conferences/10.1049/ic%7B%5C_%7D20080466.

Robles, G. et al. (Aug. 2016). “Multiple partial discharge source discrimination with multiclass support vector machines”. In: *Expert Systems with Applications* 55, pp. 417–428. ISSN: 09574174. DOI: 10.1016/j.eswa.2016.02.014. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417416300379.

Rogge-Solti, A., Aalst, W. M. P. van der, and Weske, M. (2013). “Discovering stochastic petri nets with arbitrary delay distributions from event logs”. In: *International Conference on Business Process Management*. Springer, pp. 15–27.

Ross, S. M. (1996). *Stochastic processes. 1996*.

Rubino, G. and Tuffin, B., eds. (Mar. 2009). *Rare Event Simulation using Monte Carlo Methods*. Chichester, UK: John Wiley & Sons, Ltd. ISBN: 9780470745403. DOI: 10.1002/9780470745403. URL: http://doi.wiley.com/10.1002/9780470745403.

Rubinstein, R. Y. and Kroese, D. P. (2016). *Simulation and the Monte Carlo method*. John Wiley & Sons.

Ruijters, E. and Stoelinga, M. (2015). “Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools”. In: *Computer Science Review* 15-16.October, pp. 29–62. ISSN: 15740137. DOI: 10.1016/j.cosrev.2015.03.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S1574013715000027.

Sampedro, C. et al. (2019). “Deep Learning-Based System for Automatic Recognition and Diagnosis of Electrical Insulator Strings”. In: *IEEE Access* 7, pp. 101283–101308. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2931144. URL: https://ieeexplore.ieee.org/document/8772209/.

Sánchez-Silva, M. and Klutke, G.-A. (2016a). *Reliability and Life-Cycle Analysis of Deteriorating Systems*. Springer Series in Reliability Engineering. Cham: Springer International Publishing. ISBN: 978-3-319-20945-6. DOI: 10.1007/978-3-319-20946-3. URL: http://link.springer.com/10.1007/978-3-319-20946-3.

— (2016b). *Reliability and life-cycle analysis of deteriorating systems*. Springer.

Sanders, W. H. and Meyer, J. F. (1991a). “Reduced Base Model Construction Methods for Stochastic Activity Networks”. In: *IEEE Journal on Selected Areas in Communications* 9.1, pp. 25–36. ISSN: 07338716. DOI: 10.1109/49.64901.

— (2001). “Stochastic activity networks: Formal definitions and concepts”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 2090.9975019, pp. 315–343. ISSN: 16113349.

Sanders, W. H. and Meyer, J. F. (1991b). “A unified approach for specifying measures of performance, dependability and performability”. In: *Dependable Computing for Critical Applications*. Springer, pp. 215–237.

Sanders, W. H. (1988). “Construction and solution of performability models based on stochastic activity networks”. PhD thesis. University of Michigan.

Sankararaman, S. (2015). “Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction”. In: *Mechanical Systems and Signal Processing* 52-53.1, pp. 228–247. ISSN: 08883270. DOI: 10.1016/j.ymssp.2014.05.029. URL: https://linkinghub.elsevier.com/retrieve/pii/S0888327014002052.

Sanniyati, C. N. et al. (Aug. 2016). “Water tree in polymeric cables: a review”. In: *Malaysian Journal of Fundamental and Applied Sciences* 12.1. ISSN: 2289-599X. DOI: 10.11113/mjfas.v11n4.402. URL: https://mjfas.utm.my/index.php/mjfas/article/view/402.

Savadjiev, K. and Farzaneh, M. (Apr. 2004). “Modeling of Icing and Ice Shedding on Overhead Power Lines Based on Statistical Analysis of Meteorological Data”. In: *IEEE Trans-*

actions on Power Delivery 19.2, pp. 715–721. issn: 0885-8977. doi: 10.1109/TPWRD.2003.822527. url: http://ieeexplore.ieee.org/document/1278431/.

Schryver, C. de et al. (Dec. 2010). “A New Hardware Efficient Inversion Based Random Number Generator for Non-uniform Distributions”. In: *2010 International Conference on Reconfigurable Computing and FPGAs*. IEEE, pp. 190–195. isbn: 978-1-4244-9523-8. doi: 10.1109/ReConFig.2010.20. url: http://ieeexplore.ieee.org/document/5695304/.

Seals, R. C. and Whapshott, G. F. (1997). *Programmable Logic: PLDs and FPGAs*. London: Macmillan Education UK. isbn: 978-1-349-14005-3. doi: 10.1007/978-1-349-14003-9. url: http://link.springer.com/10.1007/978-1-349-14003-9.

Shafiq, M. et al. (Mar. 2014). “Effect of geometrical parameters on high frequency performance of Rogowski coil for partial discharge measurements”. In: *Measurement* 49.1, pp. 126–137. issn: 02632241. doi: 10.1016/j.measurement.2013.11.048. url: https://linkinghub.elsevier.com/retrieve/pii/S0263224113006015.

Shao, Z. et al. (July 2018). “Dielectric Response of Cross-Linked Polyethylene (XLPE) Cable Insulation Material to Radiation and Thermal Aging”. In: *2018 IEEE 2nd International Conference on Dielectrics (ICD)*. IEEE, pp. 1–4. isbn: 978-1-5386-6389-9. doi: 10.1109/ICD.2018.8514664. url: https://ieeexplore.ieee.org/document/8514664/.

Shoar, S. et al. (2017). “Fast Fault Tree Analysis for Hybrid Uncertainties Using Stochastic Logic Implemented on Field-Programmable Gate Arrays: An Application in Quantitative Assessment and mitigation of Welding Defects Risk”. In: *Quality and Reliability Engineering International* 33.7, pp. 1367–1385. issn: 07488017. doi: 10.1002/qre.2110. url: http://doi.wiley.com/10.1002/qre.2110.

Short, K. L. (2009). *VHDL for engineers*, xxxii, 685 p. isbn: 9780131424784.

Shumaker, B. D. et al. (July 2013). “Remaining useful life estimation of electric cables in nuclear power plants”. In: *Chemical Engineering Transactions*. Vol. 33, pp. 877–882. doi: 10.3303/CET1333147. url: https://www.cetjournal.it/index.php/cet/article/view/CET1333147.

Siddiqui, B. A. et al. (2016). “A versatile solution for continuous on-line PD monitoring”. In: *Proceedings of the 2015 IEEE Innovative Smart Grid Technologies - Asia, ISGT ASIA 2015* 60270, pp. 1–6. doi: 10.1109/ISGT-Asia.2015.7387157.

Sihite, J. F. (2013). “Failure Analysis of Power Transformer Based on Fault Tree Analysis”. PhD thesis. Kyoto University. doi: 10.14989/doctor.k17885.

Simmons, K. L. et al. (2013). “Determining Remaining Useful Life of Aging Cables in Nuclear Power Plants–Interim Study FY13”. In: *Pacific Northwest National Laboratory Richland, Washington, The US Department of Energy, PNNL 22812*.

Soderstrand, M. A., Sema, A. E. de la, and Loomis, H. H. (1995). “New approach to clustered look-ahead pipelined IIR digital filters”. In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 42.4, pp. 269–274.

Song, H. and Schnieder, E. (Dec. 2018). “Evaluating Fault Tree by means of Colored Petri nets to analyze the railway system dependability”. In: *Safety Science* 110, pp. 313–323. issn: 09257535. doi: 10.1016/j.ssci.2018.08.017. url: https://linkinghub.elsevier.com/retrieve/pii/S0925753517320143.

Soto, E. and Pereira, M. (2005). “Implementing a Petri net specification in a FPGA using VHDL”. In: *Design of embedded control systems*. Springer, pp. 167–174.

Srinivasan, A., Mascagni, M., and Ceperley, D. (2003). “Testing parallel random number generators”. In: *Parallel Computing* 29.1, pp. 69–94. issn: 01678191. doi: 10.1016/S0167-8191(02)00163-1.

Stojanovic, M., Tasic, D., and Ristic, A. (2013). “Cyclic current ratings of single-core XLPE cables with respect to designed life time”. In: *Przeglad Elektrotechniczny* 89.5, pp. 152–156. issn: 0033-2097.

Suwanasri, C. et al. (July 2014). “Failure rate analysis of circuit breaker and its preventive maintenance application”. In: *2014 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. IEEE. IEEE, pp. 1–6. isbn: 978-1-4799-3561-1. doi: `10.1109/PMAPS.2014.6960623`. url: `http://ieeexplore.ieee.org/document/6960623/`.

Takahashi, T., Takahashi, T., and Okamoto, T. (2008). “Study on insulation diagnosis for XLPE cables using dumping oscillating high voltage”. In: *Condition Monitoring and Diagnosis, 2008. CMD 2008. International Conference on*. IEEE, pp. 1254–1257.

Taussky, O. and Knuth, D. E. (2006). “The Art of Computer Programming. Volume 2: Seminumerical Algorithms.” In: *The American Mathematical Monthly*. issn: 00029890. doi: `10.2307/2317055`.

Teh, J., Lai, C.-M. M., and Cheng, Y.-H. H. (2017). “Impact of the real-time thermal loading on the bulk electric system reliability”. In: *IEEE Transactions on Reliability* 66.4, pp. 1110–1119. issn: 00189529. doi: `10.1109/TR.2017.2740158`.

The Möbius Tool (2017). url: `https://www.mobius.illinois.edu` (visited on 10/15/2017).

Thomas, D. B., Luk, W., et al. (Nov. 2007). “Gaussian random number generators”. In: *ACM Computing Surveys* 39.4, 11–es. issn: 03600300. doi: `10.1145/1287620.1287622`. url: `http://portal.acm.org/citation.cfm?doid=1287620.1287622`.

Thomas, D. B. and Luk, W. (2006). “Efficient hardware generation of random variates with arbitrary distributions”. In: *Field-Programmable Custom Computing Machines, 2006. FCCM’06. 14th Annual IEEE Symposium on*. IEEE, pp. 57–66.

Trappey, C. V. et al. (Mar. 2014). “Data driven modeling for power transformer lifespan evaluation”. In: *Journal of Systems Science and Systems Engineering* 23.1, pp. 80–93. issn: 1004-3756. doi: `10.1007/s11518-014-5227-z`. url: `http://link.springer.com/10.1007/s11518-014-5227-z`.

Uehara, H. et al. (Oct. 2014). “Effect of porous metal electrode on water tree propagation”. In: *2014 IEEE Conference on Electrical Insulation and Dielectric Phenomena (CEIDP)*. IEEE, pp. 381–384. isbn: 978-1-4799-7525-9. doi: `10.1109/CEIDP.2014.6995732`. url: `http://ieeexplore.ieee.org/document/6995732/`.

Van Wormer, F. C. (Jan. 1955). “An Improved Approximate Technique for Calculating Cable Temperature Transients [includes discussion]”. In: *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems* 74.3, pp. 277–281. issn: 0097-2460. doi: `10.1109/AIEEPAS.1955.4499079`. url: `http://ieeexplore.ieee.org/document/4499079/`.

Volkanovski, A., Čepin, M., and Mavko, B. (June 2009). “Application of the fault tree analysis for assessment of power system reliability”. In: *Reliability Engineering & System Safety* 94.6, pp. 1116–1127. issn: 09518320. doi: `10.1016/j.ress.2009.01.004`. url: `https://linkinghub.elsevier.com/retrieve/pii/S0951832009000210`.

Volovoi, V. et al. (n.d.). “Modeling the reliability of distribution systems using Petri nets”. In: *2004 11th International Conference on Harmonics and Quality of Power (IEEE Cat. No.04EX951)*. IEEE, pp. 567–572. isbn: 0-7803-8746-5. doi: `10.1109/ICHQP.2004.1409416`. url: `http://ieeexplore.ieee.org/document/1409416/`.

Walker, M. and Papadopoulos, Y. (Oct. 2009). “Qualitative temporal analysis: Towards a full implementation of the Fault Tree Handbook”. In: *Control Engineering Practice* 17.10, pp. 1115–1125. issn: 09670661. doi: `10.1016/j.conengprac.2008.10.003`. url: `https://linkinghub.elsevier.com/retrieve/pii/S096706610800186X`.

Wallace, P. A. et al. (2009). “Failure Modes of Underground MV Cables: Electrical and Thermal Modelling”. In: *COMSOL Conference 2009*. url: `http://www.ch.comsol.com/papers/6783/download/Wallace.pdf`.

Wang, P. and Vachtsevanos, G. (2001). “Fault prognostics using dynamic wavelet neural networks”. In: *AI EDAM* 15.4, pp. 349–365.

Wei, C., Chengshu, L., and Xin, S. (2004). “FPGA implementation of universal random number generator”. In: *Signal Processing, 2004. Proceedings. ICSP'04. 2004 7th International Conference on*. Vol. 1. IEEE, pp. 495–498.

Whitt, W. (1980). “Continuity of Generalized Semi-Markov Processes.” In: *Mathematics of Operations Research* 5.4, pp. 494–501. ISSN: 0364765X. DOI: 10.1287/moor.5.4.494.

Wijayarathna, P. G. and Maekawa, M. (2000). “Extending fault trees with an AND-THEN gate”. In: *Proceedings of the International Symposium on Software Reliability Engineering, ISSRE*, pp. 283–292. ISSN: 10719458.

Wild, F. D. E. et al. (2015). *A Guide for Rating Calculations of Insulated Cables, Working group B1.35.* December, pp. 1–6. ISBN: 9782858733439.

Wilkins, E. W. C. (1956). “Cumulative damage in fatigue”. In: *Colloquium on Fatigue/Colloque de Fatigue/Kolloquium über Ermüdungsfestigkeit*. Springer, pp. 321–332.

Willis, H. L. (2004). *Power distribution planning reference book*. CRC press.

Wu, Y.-F. and Lewins, J. D. (1992). “Monte Carlo studies of engineering system reliability”. In: *Annals of Nuclear Energy* 19.10-12, pp. 825–859.

Xiao, L. et al. (Oct. 2017). “A Fault Diagnosis Method of Insulator String Based on Infrared Image Feature Extraction and Probabilistic Neural Network”. In: *2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA)*. IEEE, pp. 80–85. ISBN: 978-1-5386-1230-9. DOI: 10.1109/ICICTA.2017.25. URL: http://ieeexplore.ieee.org/document/8089908/.

Xilinx Inc. (2018a). *Vivado High-Level Synthesis*. URL: https://www.xilinx.com/products/design-tools/vivado/integration/esl-design.html (visited on 12/29/2018).

— (2018b). *Zynq UltraScale+ MPSoC Data Sheet: DC and AC Switching Characteristics*.

Yianni, P. C. et al. (Feb. 2018). “Accelerating Petri-Net simulations using NVIDIA Graphics Processing Units”. In: *European Journal of Operational Research* 265.1, pp. 361–371. ISSN: 03772217. DOI: 10.1016/j.ejor.2017.06.068. URL: https://linkinghub.elsevier.com/retrieve/pii/S0377221717306276.

Yilmaz, G. and Karlik, S. E. (Jan. 2006). “A distributed optical fiber sensor for temperature detection in power cables”. In: *Sensors and Actuators A: Physical* 125.2, pp. 148–155. ISSN: 09244247. DOI: 10.1016/j.sna.2005.06.024. URL: https://linkinghub.elsevier.com/retrieve/pii/S0924424705003766.

Zhang, D. et al. (June 2011). “Evaluating Condition Index and Its Probability Distribution Using Monitored Data of Circuit Breaker”. In: *Electric Power Components and Systems* 39.10, pp. 965–978. ISSN: 1532-5008. DOI: 10.1080/15325008.2011.552091. URL: http://www.tandfonline.com/doi/abs/10.1080/15325008.2011.552091.

Zhang, G. et al. (2005). “Ziggurat-based hardware gaussian random number generator”. In: *Proceedings - 2005 International Conference on Field Programmable Logic and Applications, FPL*. Vol. 2005. IEEE, pp. 275–280. ISBN: 0780393627. DOI: 10.1109/FPL.2005.1515734.

Zhang, X. and Pisu, P. (2012). “An unscented kalman filter based approach for the health-monitoring and prognostics of a polymer electrolyte membrane fuel cell”. In: *Annual conference of the prognostics and health management society*. Vol. 3, pp. 1–9. ISBN: 9781936263059.

Zhang, Z., Assala, P. D. S., and Wu, L. (Oct. 2018). “Residual life assessment of 110 kV XLPE cable”. In: *Electric Power Systems Research* 163, pp. 572–580. ISSN: 03787796. DOI: 10.1016/j.epsr.2017.12.027. URL: https://linkinghub.elsevier.com/retrieve/pii/S0378779617305060.

Zhong, J. et al. (Nov. 2015). “Incorporating a Condition Monitoring Based Aging Failure Model of a Circuit Breaker in Substation Reliability Assessment”. In: *IEEE Transactions on Power Systems* 30.6, pp. 3407–3415. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2014.2387334. URL: http://ieeexplore.ieee.org/document/7006780/.

Zhou, Y. et al. (2009). “Asset life prediction using multiple degradation indicators and lifetime data: A Gamma-based state space model approach”. In: *Proceedings of 2009 8th International Conference on Reliability, Maintainability and Safety, ICRMS 2009*, pp. 445–449. doi: 10.1109/ICRMS.2009.5270153.

Zhu, P. et al. (Nov. 2017). “A stochastic analysis of competing failures with propagation effects in functional dependency gates”. In: *IISE Transactions* 49.11, pp. 1050–1064. issn: 2472-5854. doi: 10.1080/24725854.2017.1342056. url: https://www.tandfonline.com/doi/full/10.1080/24725854.2017.1342056.

Zhurkov, S. N. (1984). “Kinetic concept of the strength of solids”. In: *International Journal of Fracture* 26.4, pp. 295–307.

Zio, E. (2013). *The Monte Carlo Simulation Method for System Reliability and Risk Analysis*. Springer Series in Reliability Engineering. London: Springer London. isbn: 978-1-4471-4587-5. doi: 10.1007/978-1-4471-4588-2. url: http://link.springer.com/10.1007/978-1-4471-4588-2.

— (2016). “Some challenges and opportunities in reliability engineering”. In: *IEEE Transactions on Reliability* 65.4, pp. 1769–1782. issn: 00189529. doi: 10.1109/TR.2016.2591504.

Zio, E. and Di Maio, F. (Jan. 2010). “A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system”. In: *Reliability Engineering & System Safety* 95.1, pp. 49–57. issn: 09518320. doi: 10.1016/j.ress.2009.08.001. url: https://linkinghub.elsevier.com/retrieve/pii/S0951832009002051.