ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

GRADO EN INGENIERÍA ELECTROMECÁNICA

INTEGRATION OF DYNAMIC THERMAL RATING MODEL OF TRANSMISSION CAPACITY TO POWER SYSTEMS OPERATION

ESPECIALIDAD ELÉCTRICA

Autor: José María Sánchez-Ventura del Águila
Director: Rachid Cherkaoui

Madrid
July 2014

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN ACCESO *ABIERTO* (*RESTRINGIDO*) DE DOCUMENTACIÓN

1º. Declaración de la autoría y acreditación de la misma.

El autor D. José María Sánchez-Ventura del Águila, como estudiante de Grado en Ingeniería Electromecánica de la UNIVERSIDAD PONTIFICIA COMILLAS (COMILLAS), **DECLARA**

que es el titular de los derechos de propiedad intelectual, objeto de la presente cesión, en relación con el Proyecto de Fin de Grado titulado Integration of dynamic thermal rating model of transmission capacity to Power Systems operation, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual como titular único o cotitular de la obra.

En caso de ser cotitular, el autor (firmante) declara asimismo que cuenta con el consentimiento de los restantes titulares para hacer la presente cesión. En caso de previa cesión a terceros de derechos de explotación de la obra, el autor declara que tiene la oportuna autorización de dichos titulares de derechos a los fines de esta cesión o bien que retiene la facultad de ceder estos derechos en la forma prevista en la presente cesión y así lo acredita.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad y hacer posible su utilización de *forma libre y gratuita* (*con las limitaciones que más adelante se detallan*) por todos los usuarios del repositorio y del portal e-ciencia, el autor **CEDE** a la Universidad Pontificia Comillas de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución, de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra (a) del apartado siguiente.

3º. Condiciones de la cesión.

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia, el repositorio institucional podrá:

(a) Transformarla para adaptarla a cualquier tecnología susceptible de incorporarla a internet; realizar adaptaciones para hacer posible la utilización de la obra en formatos electrónicos, así como incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.

(b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato. .

(c) Comunicarla y ponerla a disposición del público a través de un archivo abierto institucional, accesible de modo libre y gratuito a través de internet.[1]

(d) Distribuir copias electrónicas de la obra a los usuarios en un soporte digital. [2]

4º. Derechos del autor.

El autor, en tanto que titular de una obra que cede con carácter no exclusivo a la Universidad por medio de su registro en el Repositorio Institucional tiene derecho a:

a) A que la Universidad identifique claramente su nombre como el autor o propietario de los derechos del documento.

b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.

c) Solicitar la retirada de la obra del repositorio por causa justificada. A tal fin deberá ponerse en contacto con el vicerrector/a de investigación (curiarte@rec.upcomillas.es).

d) Autorizar expresamente a COMILLAS para, en su caso, realizar los trámites necesarios para la obtención del ISBN.

d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.

b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.

---

[1] En el supuesto de que el autor opte por el acceso restringido, este apartado quedaría redactado en los siguientes términos:

(c) Comunicarla y ponerla a disposición del público a través de un archivo institucional, accesible de modo restringido, en los términos previstos en el Reglamento del Repositorio Institucional

[2] En el supuesto de que el autor opte por el acceso restringido, este apartado quedaría eliminado.

c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

a) Deberes del repositorio Institucional:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.

- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.

- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.

b) Derechos que se reserva el Repositorio institucional respecto de las obras en él registradas:

- retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 20 de julio de 2014

ACEPTA

Fdo……………………………………………………………

Autorizada la entrega del proyecto al alumno:

José María Sánchez-Ventura del Águila

EL DIRECTOR DEL PROYECTOR

Rachid Cherkaoui

Fdo: Fecha: 03/07/2014

Vº Bº DEL COORDINADOR DE PROYECTOS

Fernando de Cuadra

Fdo: Fecha: / /

Summary of the project

Introduction

In the last decades, several studies have demonstrated that static thermal rating uses to be too conservative and misses important power capacity that could be better exploited with a good dynamic thermal rating. That is why many governments with the aid of electrical companies, firms and universities have been doing an important effort during the last years in order to study how to take advantage of it. In fact, as it has already tested in different countries such us United States, United Kingdom and Sweden, the integration of dynamic thermal rating can clearly imply important performance, safety and also economical benefits to the electrical system.

This has been the main motivation for developing this project: to study different ways of making the most of dynamic thermal rating and show some potential benefits through its integration to Power System Operation, especially to Security Analysis.

Objective

We were looking to create a versatile tool by the way that any electrical network data (previously adapted to MATLAB format) as well as meteorological conditions (ambient temperature, wind speed and solar heat) could be uploaded to run deterministic (real-time) and probabilistic (forecasting) evaluation calculating all the line temperatures for the base case and for all the N-1 contingencies.

Design: Incorporation of dynamic thermal rating to power systems security

First of all, a thermal evaluation of overhead transmissions lines was done. The standard IEEE 738 method for calculating the current-temperature relationship of bare over-head has been applied. Thus, the first steps were devoted to find an efficient computational way of solving the heat balance equation in steady state, which is no linear and can be seen below:

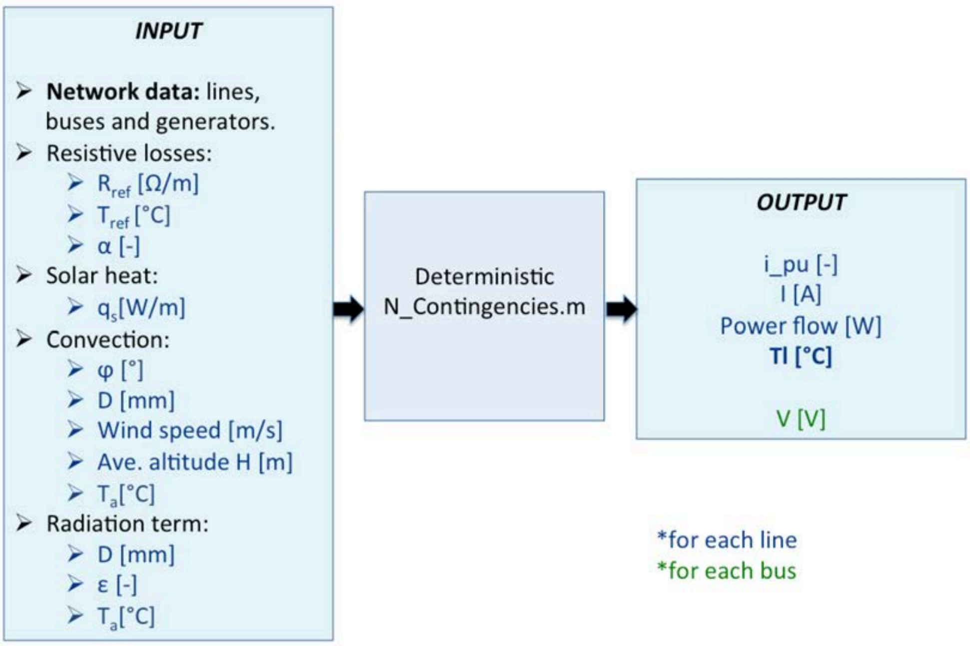
$$0 = I^2 \cdot R(T_l) + q_s - q_c(T_l) - q_r(T_l)$$

Secondly, for security evaluation, the load flow was run as like as all the N-1 contingencies of the system with MATPOWER, a MATLAB package for solving power flow and optimal power flow problems. This is absolutely necessary to be able to calculate the line current of each case and contingency, which is actually the key variable that relates electrical and thermal equations.

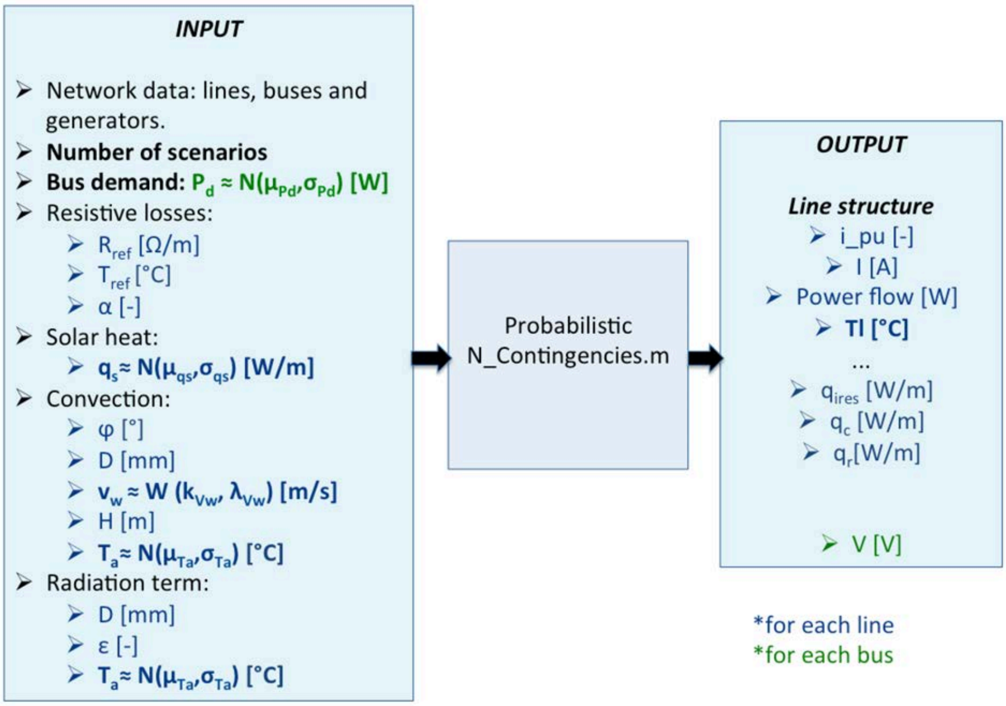
In addition, at one point, a wind farm has been modeled and incorporated to the network substituting one of the existing generators so that its production output varies in function of the input wind speed making the system to adapt to the power unbalances.

On the other hand, two different kinds of approaches were implemented:

- *Deterministic evaluation:* especially useful when dealing with real-time data acquisition and uncertainty does not represent a big matter. The input variables (weather conditions) are unique and concrete, so the desired output variables (line temperatures) are unique too for each case.



- *Probabilistic evaluation:* in order to deal with uncertainties coming from weather conditions and demand nodes another kind of approach was developed. These different types of uncertainties were firstly modeled with probabilistic distributions (normal and weibull) by the way that their corresponding parameters could well define them. Afterwards, accurate samples obtained though Latin Hypercube Sampling method were generated from these distributions to run a large number of simulations and to be able to analyze uncertainties' ranges.



Therefore the main steps in order followed by both .m programs (DeterministicN_Contigencies and ProbabilisticN_Contingencies) are:

1) Upload to MATLAB the network case (IEEE test cases or any other real one converted to MATLAB code).

2) Upload/fix: the weather parameters of each line (ambient temperature, wind speed, solar heat) and line conductors specifications (T_{ref}, R_{ref}, D).

   *Only Probabilistic: specify the value of N (number of simulations) so that it generates N samples for each variable through the given probabilistic parameters.

3) Run the load flow through MATPOWER for the base case and for all contingencies (1 time deterministic/ N times probabilistic).

4) Calculation (with the difference between the two connected buses' voltages divided by the line impedance) and storage of line currents in per units and after that in real values of each line for the base case, all contingencies (and each scenario when it is probabilistic).

5) Resolution of the non-linear heat balance equation in order to calculate the temperature of each line for the base case, all contingencies (and each scenario when it is probabilistic).

6) Verification of non-maximum temperature exceeded at any line or case and results analysis.

Conclusions and next steps

Interesting graphs and conclusions have been obtained representing the comparison between the static and dynamic thermal rating at different weather conditions and also studying the most critical contingencies. Results show indeed how this margin between the real and conservative rating could be highly exploited.

Apart from Security Analysis, dynamic thermal rating may be also useful for: system planning, contract design and market analysis. Therefore, there is still quite a lot of work to do regarding these points.

For useful probabilistic evaluation, good weather and load forecast becomes absolutely necessary in order to obtain realistic line temperature forecast. Therefore, an important effort on collecting real data of them would be required because that is on what forecasting should be based.

Finally, as it has been mentioned, a versatile tool was tried to be developed by the way that any network with the necessary state data (previously format adapted to MATLAB) could be uploaded to calculate all the line temperatures for the base case and for all contingencies on a deterministic (applicable for real time) or on a probabilistic way (forecasting and analysis).

Resumen del proyecto

Introducción

En la últimas décadas, numerosos estudios han demostrado que los parámetros térmicos estáticos de las líneas tienden a ser excesivamente conservadores y no aprovechan una parte importante de la capacidad de transferir potencia de las líneas que podría ser mejor explotadas con unos buenos parámetros térmicos dinámicos en su lugar. Esta es la razón por la cual muchos gobiernos con la ayuda de compañías eléctricas, empresas y universidades han estado haciendo un importante esfuerzo estos los últimos años para sacar un mayor partido del mismo. De hecho, tal y como ha sido probado en diferentes países, como por ejemplo en Estados Unidos, Reino Unido y Suecia, la integración de los parámetros dinámicos de línea puede presentar importantes mejoras de rendimiento, seguridad y ahorro económico para el sistema eléctrico.

Ésta ha sido principalmente la motivación para desarrollar este proyecto: estudiar formas de sacar mayor provecho de los parámetros térmicos dinámicos y mostrar sus ventajas al integrarlo al sistema eléctrico de potencia; en concreto para el análisis de seguridad.

Objetivo

Se pretende crear una herramienta versátil y polivalente de manera que introduciendo la información correspondiente de cualquier red eléctrica (previamente adaptada al lenguaje del MATLAB) al igual que las condiciones meteorológicas (temperatura ambiente, velocidad del viento y radiación solar) sea posible llevar a cabo simulaciones deterministas (tiempo real) o probabilísticas (para forecast y previsión) obteniendo las temperaturas de cada línea para el caso base y también para todas las contingencias simples (N-1).

Diseño: Incorporación de los parámetros térmicos dinámicos al análisis de seguridad.

En primer lugar se realizó una análisis térmico de los conductores aéreos de transmisión. Para ello, se aplicó el método mostrado en el IEEE 738 para el cálculo de la relación corriente-temperatura de las líneas aéreas. De esta forma, los primeros pasos consistieron en solucionar eficientemente en términos de computación la ecuación de equilibrio térmico en estado estacionario, la cual es no lineal y se muestra a continuación:

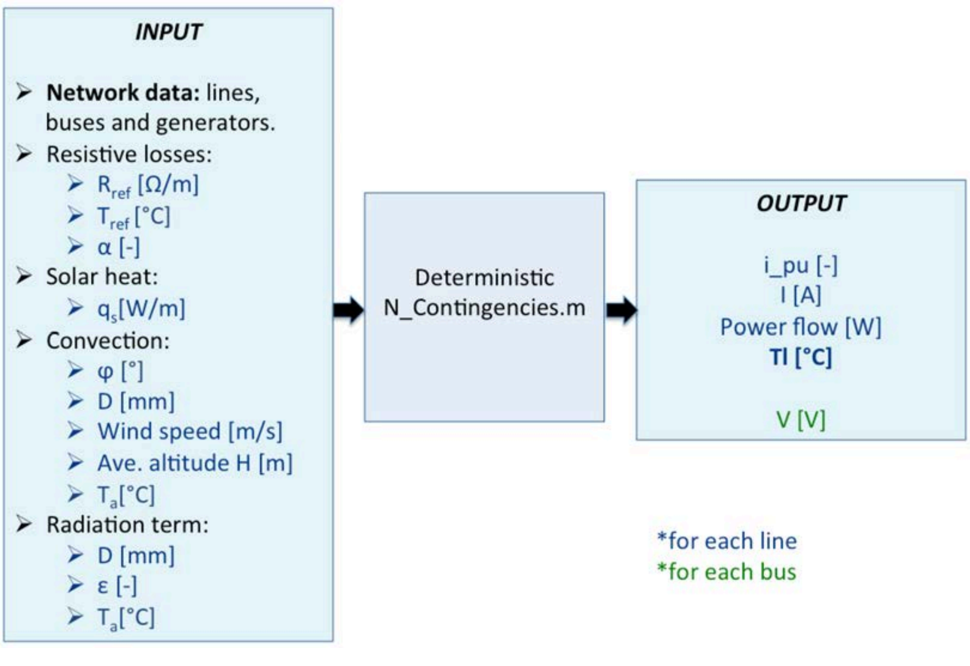
$$0 = I^2 \cdot R(T_l) + q_s - q_c(T_l) - q_r(T_l)$$

En segundo lugar, para el análisis de seguridad, se soluciona el flujo de cargas así como todas las contingencias simples (N-1) con MATPOWER, una aplicación asociada a MATLAB para resolver flujos de carga y problemas de flujos óptimos de carga. Este paso es imprescindible para calcular las corrientes de línea en cada caso y contingencia; siendo la intensidad la variable clave que relaciona las ecuaciones eléctrica y térmica.

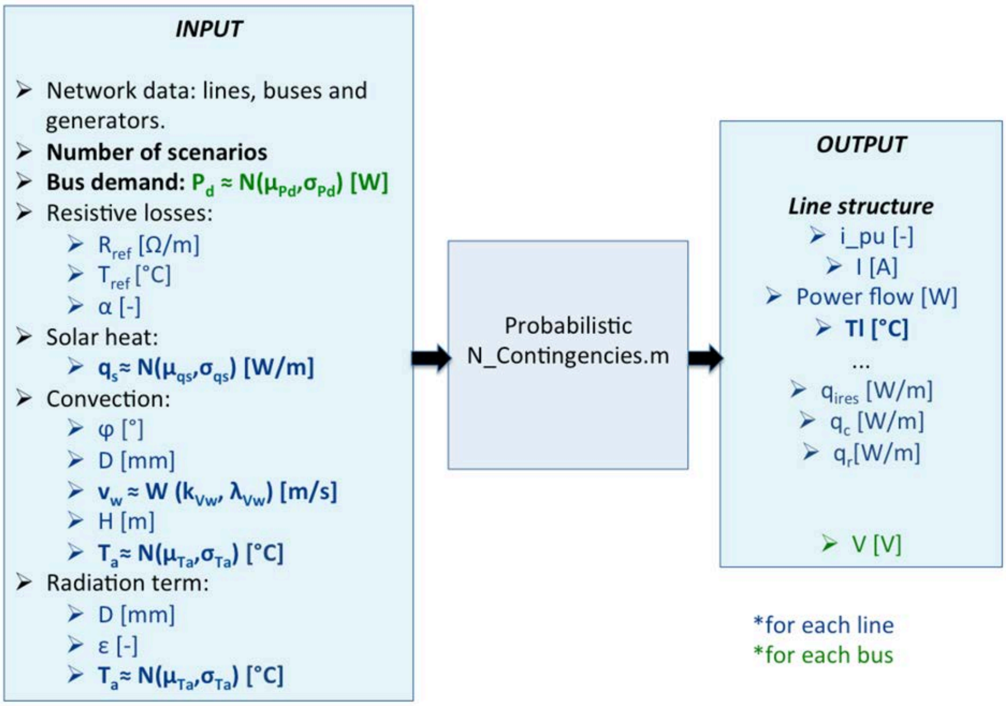
Además, un parque eólico es modelado e incorporado a la red sustituyendo uno de los generadores existentes. De esta manera, se establece que la producción del parque variará en función de la velocidad del viento inyectada, esto generará desequilibrios que el sistema deberá compensar.

Por otra lado, se desarrollaron dos posibles diseños:

- *Evaluación determinista:* especialmente útil cuando se aplica en tiempo real y la incertidumbre no representa un problema. Las variables de entrada (condiciones meteorológicas) son únicas y concretas; de forma que las variables de salida (temperatura de la línea) también lo son.



- *Evaluación probabilística:* otro tipo de diseño se aplica para poder analizar las incertidumbres procedentes de las condiciones meteorológicas y de la demanda en los nodos. Estas incertidumbres son en primer lugar modeladas por medio de distribuciones estadísticas (normal y weibull) de forma que los correspondientes parámetros (media y varianza o forma y escala) puedan definirlas. Posteriormente, se emplea el método Latin Hypercube para generar muestras precisas y poder llevar a cabo las simulaciones.



Por tanto, a continuación los principales etapas del proceso que siguen los programas diseñados (DeterministicN_Contigencies.m y ProbabilisticN_Contingencies.m) son:

1) Cargar en MATLAB la información de la red en estudio (bien uno de los casos de prueba IEEE o cualquier otra red real lista para trabajar en MATLAB).

2) Cargar/fijar: los parámetros meteorológicos de cada línea (temperatura ambiente, velocidad del viento y radiación solar) y las especificaciones técnicas de los conductores (T_{ref}, R_{ref}, D, etc).

*Sólo para el estudio probabilístico: especificar el valor de N (número de simulaciones) de forma que el programa genera N muestras para cada variable a través de los parámetros de las distribuciones fijados.

3) Correr el flujo de cargas con MATPOWER para el caso base y para todas las contingencias simples (Una vez únicamente para determinista/ N veces para la simulación probabilística).

4) Cálculo (por medio de la diferencia entre los voltajes de los nudos dividido entre la impedancia de la línea) y almacenamiento de las corrientes de la línea en p.u. y tras esto en valores reales para el caso base y todas las contingencias simples (y para cada escenario cuando es probabilística).

5) Resolución de la ecuación no lineal de equilibrio térmico para obtener finalmente las temperaturas de cada línea para el caso base y todas las contingencias (y para cada escenario cuando es probabilística).

6) Comprobación de que no se excede en ningún caso la temperatura máxima y análisis de resultados.

Conclusiones y próximos pasos

Interesantes gráficas y conclusiones han podido obtenerse a partir de la representación comparativa entre los parámetros térmicos estáticos y dinámicos para diferentes condiciones meteorológicas y observando las contingencias más críticas. Los resultados reflejan en efecto que existe un margen importante a explotar entre los parámetros estáticos que se fijan y los reales.

Además de para el análisis de seguridad, los parámetros térmicos dinámicos podrían resultar especialmente útiles en otros campos como: planificación de sistema, diseño de contratos y análisis de mercados. Por lo tanto, aún queda un extenso campo de trabajo del que se podrían obtener resultados muy positivos para la mejora del sistema.

Para llevar a cabo una precisa evaluación probabilística con previsión realista de temperaturas de línea, es absolutamente imprescindible contar con previsiones meteorológicas y de demanda de calidad y bien fundamentadas. Consecuentemente, es muy importante realizar un esfuerzo importante en la recopilación de datos, ya que es la pieza sobre la cual se ha fundamentar la previsión.

Para acabar, tal y como ya ha sido mencionado en los objetivos, se ha pretendido desarrollar una herramienta polivalente de forma con los datos de funcionamiento de que cualquier red con el formato adecuado se puedan calcular todas las temperaturas de las líneas tanto para el caso base como para las contingencias simples de forma determinista (aplicable en tiempo real) o probabilística (para previsión y análisis).

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

GRADO EN INGENIERÍA ELECTROMECÁNICA

INTEGRATION OF DYNAMIC THERMAL RATING MODEL OF TRANSMISSION CAPACITY TO POWER SYSTEMS OPERATION

ESPECIALIDAD ELÉCTRICA

Autor: José María Sánchez-Ventura del Águila
Director: Rachid Cherkaoui

Madrid
July 2014

Abstract

This paper shows a way of integrating thermal rating to Power System Operation and especially for Security Evaluation. Probabilistic methods have been used to model weather (ambient temperature, wind speed and solar heat) and demand uncertainties; also a variable speed wind farm has been incorporated to the network. Case studies results have been simulated through Latin Hypercube Sampling method. To do so, a MATLAB file have been created that permits uploading a network data (generators, lines and loads) to run the deterministic and/or probabilistic simulation with the previously fixed meteorological conditions to calculate the temperature of each line for the base case and for all N-1 contingencies.

Contents

List of figures

List of tables

Glossary

HBE	Heat Balance Equation
IEEE 738	Standard for Calculating the Current-Temperature of Bare Overhead Conductors
LHS	Latin Hypercube Sampling
MCS	Monte Carlo Sampling
LHSMC	Latin Hypercube Sampling Monte Carlo
AC	Alternating Current
DC	Direct Current
OPF	Optimal Power Fload

1 Introduction

During the last years, it has been demonstrated that static rating uses to be too conservative and misses important power capacity that could be better exploited with a good thermal rating without risking systems security but improving its performance.

The aim of this project has been to study different ways of making the most of dynamic thermal rating and show potential benefits through its integration to Power System Operation, especially to Security Analysis. To do so, the structure of this document can be divided into four parts that are briefly summarized over the following lines.

Firstly, introductive sections 2 and 3 where key concepts related with the Heat Balance Equation (HBE) as like as those with Security Evaluation of Power System Evaluation are defined. Section 2 is based on IEEE 738 and reflects the physical equations that rules the rates of thermal energy of the conductor. On the other hand, at section 3, apart from giving a short and overall revision of the load flow and the N-1 contingency analysis, it is explained how this have been implemented by the Power System Simulation Package of MATLAB that is MATPOWER, the simulation tool used to run all the load flow and contingencies. Also, it is described how the system has been designed to respond to power unbalances; at the end a wind farm has been modeled and incorporated to the network substituting one of the existing generators.

Secondly, the practical resolution procedure of the HBE is shown at section 4, moreover, the principal steps always followed at this work to stir in it to the Security Analysis. Deterministic evaluation is done at the end of this chapter; this is actually the basis that will lead to next sections probabilistic evaluation.

Third part corresponds to section 0 where uncertainties coming from weather parameters and demand are modeled. Stochastic or probabilistic evaluation is performed through scenario construction methodology: Latin Hyper Cube sampling; this method is explained there too.

Finally, section 0 is entirely devoted to results presentation from simulations and final conclusions are summed up at section 0.

In addition, appendix 1 and appendix 2 have been included to provide additional information and for better understanding of: wind farm model output and MATLAB stored results structure (defined as “line.mat”).

2 Thermal evaluation of overhead transmission lines

Heat Balance Equation (HBE)

The Heat Balance Equation (HBE) expresses the rates at which thermal energy enters, gets stored and leaves in the electric conductor.

$$m \cdot C_p \cdot \frac{dT_l(t)}{dt} = I^2 \cdot R(T_l(t)) + q_s(t) - q_c(T_l(t)) - q_r(T_l(t))$$

Equation 2.1. Heat balance equation

Equation 2.1 as like as the following tables and equations were all taken from IEEE 738. In Table 2.1 all variables that will appear at this section are specified with a brief description and their corresponding SI units:

Symbol	Description	SI units
A’	Projected area of conductor per unit of length	m^2/m
C	Solar azimuth constant	degrees
D	Conductor diameter	mm
H	Elevation of conductor above sea level	m
H_c	Altitude of sun	degrees
I	Conductor current	C
K_{angle}	Wind direction factor	-
K_{solar}	Solar altitude correction factor	-
k_f	Thermal conductivity of air at temperature T_f	W/(m·°C)
Lat	Degrees of latitude	degrees
$m \cdot C_p$	Total heat capacity of conductor	J/(m·°C)
N	Day of the year (January 21=21, February 12=43, etc.)	-
q_{cn}, q_{c1}, q_{c2}	Convected heat loss rate per unit length	W/m
q_r	Radiated heat loss rate per unit length	W/m
q_s	Heat gain rate from sun	W/m
Q_s	Total solar and sky radiated heat flux rate	W/m^2
Q_{se}	Total solar and sky radiated heat flux rate elevation corrected	W/m^2

$R(T_l)$	AC resistance of conductor at temperature T_l	Ω/m
T_a	Ambient air temperature	°C
T_l	Line temperature	°C
T_f	$(T_l+T_a)/2$	°C
V_w	Speed of air stream at conductor	m/s
Z_c	Azimuth of sun	degrees
Z_l	Azimuth of line	degrees
α_s	Solar absorptivity (0.23 to 0.91)	-
δ	Solar declination (0 to 90)	degrees
ε	Emissivity (0.23 to 0.91)	-
ϕ	Angle between wind and axis of conductor	degrees
ρ_f	Density of air	kg/m^3
θ	Effective angle of incidence of the sun's rays	degrees
μ_f	Dynamic viscosity of air	Pa·s
ω	Hours from local sun noon times 15	degrees
χ	Solar azimuth variable	-

Table 2.1.Units and identification of letter symbols. IEEE std 738-2006

In our case, to simplify calculations and as it is not the goal of this document to analyze transients; we are always going to consider steady state conditions. This is already a good method to check security analysis in normal working conditions.

Thus, the derivative term on the left from the previous equation becomes zero.

$$0 = I^2 \cdot R(T_l) + q_s - q_c(T_l) - q_r(T_l)$$

Equation 2.2. Heat balance equation in steady state

As it can be elucidated from last equation considering a positive sign criterion for entering energy in the system, the conductor in this case, and also as it can be seen on Figure 2.1, it is possible to distinguish two different heat sources:

- Internal heating due to the current the flows through the conductor, $I^2 \cdot R$. In fact, from this term we obtain the relation between the electrical and thermal equations. That is why, we will always run at first the load flow looking to obtain the currents of each line and afterwards we proceed with HBE resolution. Anyway, this will be explained later on more in detail.

  The AC resistance of the conductor varies with the temperature of line and will be calculated as follows:

$$R = R_{ref} \cdot \left(1 + \alpha \cdot \left(T_l - T_{ref}\right)\right)$$

Equation 2.3. AC resistance of a conductor.

  Of course R_{ref}, α and T_{ref} depend on the characteristics of the line. However, here all lines were assumed to have the same values: T_{ref}=20ºC, $R_{ref}=86 \cdot 10^{-6}$ Ω/m and α=0.0036 1/C.

- Solar heating due to solar radiation, q_s. The equation supplied by IEEE 738 is the following one:

$$q_s = \alpha_s \cdot Q_{se} \cdot \sin\theta \cdot A'$$

Equation 2.4

  Where:

$$\theta = arcos[\cos(H_c) \cdot \cos\,(Z_c - Z_l)]$$

Equation 2.5

  The solar altitude of the sun, H_s, is equal to:

$$H_c = \arcsin\,[\cos(Lat) \cdot \cos(\delta) \cdot \cos(\omega) + \sin(Lat) \cdot \sin\,(\delta)]$$

Equation 2.6

  As it was defined at Table 2.1, ω is the hour angle or the number of hours from noon times 15° (11 a.m. is -15° and 2 p.m. is +30°).

The solar declination, δ is shown in Equation 2.7. The argument of the sin is in degrees and it is valid for all latitudes: positive for northern hemisphere or negative for the southern hemisphere.

$$\delta = 23.4583 \cdot sin\left[\frac{284 + N}{365} \cdot 360\right]$$

Equation 2.7

The solar azimuth, Z_c:

$$Z_c = C + arctan(\chi)$$

Equation 2.8

And finally:

$$\chi = \frac{\sin(\omega)}{\sin(Lat) \cdot \cos(\omega) - \cos(Lat) \cdot \tan(\delta)}$$

Equation 2.9

At IEEE can be found some useful tables to help to calculate all these parameters. Though, at this work, we will assume that, as we know the network geographical area (latitude, solar altitude, etc.), we will directly introduce in the HBE reasonable and realistic values of q_s avoiding all these calculations.

And also the following two cooling sources:

- The convection term q_c, which is a function of the wind speed, the ambient and the conductor temperature. Now we show the non-linear equations provided by IEEE 738 [15] of the convection term.

  For any wind speed different zero, it is recommended to use the higher value between q_{c1} and q_{c2} of these two equations:

$$q_{c1} = \left[1.01 + 0.0372 \cdot \left(\frac{D \cdot \rho_f \cdot v_w}{\mu_f}\right)^{0.52}\right] \cdot k_f \cdot K_{angle} \cdot (T_l - T_a)$$

Equation 2.10. Forced convection heat loss rate at low wind [W/m]

$$q_{c2} = \left[0.0119 \cdot \left(\frac{D \cdot \rho_f \cdot v_w}{\mu_f}\right)^{0.6}\right] \cdot k_f \cdot K_{angle} \cdot (T_l - T_a)$$

Equation 2.11. Forced convection heat loss rate at high wind [W/m]

However, at natural convection what is to say: v_w=0. Then:

$$q_{cn} = 0.0205 \cdot \rho_f^{0.5} \cdot D^{0.75}(T_l - T_a)^{1.25}$$

Equation 2.12. Natural convection heat loss rate [W/m]

Standard tables and equations to calculate ρ_f, μ_f and k_f can be found at section 0, by the way, it is important to define ϕ as the wind direction and the conductor axis. In fact, K_{angle} is a function of it:

$$K_{angle} = 1.194 - \cos(\phi) + 0.194 \cdot \cos(2\phi) + 0.368 \cdot \sin(2\phi)$$

Equation 2.13. Wind direction factor [-]

- The radiation term q_r, which depends on the ambient and conductors temperature.

$$q_r = 0.0178 \cdot D \cdot \varepsilon \cdot \left[\left(\frac{T_l + 273}{100}\right)^4 - \left(\frac{T_a + 273}{100}\right)^4\right]$$

Equation 2.14. Radiated heat loss rate [W/m]

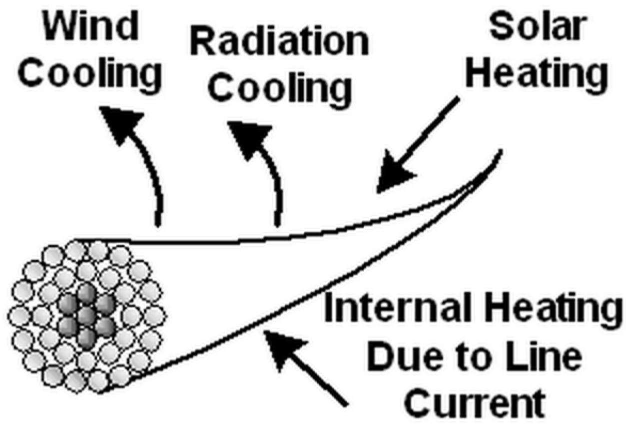


Figure 2.1. Heat terms of an electrical conductor [14]

Finally, it should be highlighted that **conductor type** is a determinant factor as it has been seen that important parameters (T_{ref}, R_{ref}, D, ε), at the previous equations, and others such as ρ_f, μ_f and k_f (look at section 0) may vary depending on the conductors' characteristics.

3 Security evaluation of power system operation

Load Flow problem

The load flow study is an essential tool in power engineering that looks to obtain all the information regarding the network operation.

At this point we are not going to dedicate any time to study the different methods that have been developed since long time ago. We will just mention some of the fundamental load flow equations and we will explain how the problem has been dealt and solved at this work.

The basic equation that rules everything is the AC nodal power balance equations. Below it is expressed as a function of the complex bus voltages and generator injections in complex matrix [8].

$$g_s(V, S_g) = S_{bus}(V) + S_d - C_g S_g = 0$$

Equation 3.1

Where: $S_{bus}(V)$ vector expresses the complex power consumed by the bus lines, S_d is the vector of complex loads at all buses; S_g is the vector of real power bus injection by all generators; C_b is a generator connection matrix that expresses where each generator is located inside the bus network. In fact, it is possible to decouple this last equation in two, either for active or for reactive power:

$$g_P(\theta, V_m, P_g) = P_{bus}(\theta, V_m) + P_d - C_g P_g = 0$$

Equation 3.2

$$g_Q(\theta, V_m, P_Q) = Q_{bus}(\theta, V_m) + Q_d - C_g Q_g = 0$$

Equation 3.3

MATPOWER was the selected tool to solve the load flow. To do so, IEEE test cases could be uploaded and implemented. Of course, any other real or not case with the adequate .m format could also be used to run the power flow. Therefore, this seems to be a very adaptable and multi-purposes tool.

In addition, MATPOWER gives the user option of choosing between a wide range of power flows (AC, DC and several Optimal Power Flows: standard AC OPF, standard DC OPF and even other extender OPF), this choice may depend on the users expected goal our desire of computation mode. Among the four different algorithms for solving the AC power flow problem (Newton's method, Fast-Decoupled XB version, Fast-Decouples BX version and Gauss-Seidel), the second one was selected.

Another good advantage that this solver has is the so practical implementation at the time that it provides an easy way to access to the results through a MATLAB structure where everything (voltages, angles, active and reactive powers, etc.) is stored.

As it was introduced at section 0, when talking about the HBE, it was said that the conductor current I represents the link between the thermal and electrical equations. Actually, calculating currents across the lines is mainly why we need to run the load flow too. However, MATPOWER results output structure does not directly calculate the currents, anyway that is not a problem because this can be easily deduced using the following equation.

So the current in per units that passes across the line that connects buses m and n can be obtained as it is shown. Here we only worry about the absolute value, not about the sign and neither the direction because it will be raised to the power of two at the HBE ($I^2 \cdot R$):

$$|i_{m-n}| = |i_{n-m}| = \left|\frac{\bar{v}_n - \bar{v}_m}{\bar{z}_{m-n}}\right| = \left|\frac{|\bar{v}_n| \cdot e^{j\theta_n} - |\bar{v}_m| \cdot e^{j\theta_m}}{r_{m-n} + jx_{m-n}}\right|$$

Equation 3.4

That is also equivalent to say:

$$|I_{m-n}| = |I_{n-m}| = |i_{m-n}| \cdot I_{base} = |i_{m-n}| \cdot \frac{S_{base}}{\sqrt{3} \cdot U_{base}} = \left|\frac{|\bar{V}_n| \cdot e^{j\theta_n} - |\bar{V}_m| \cdot e^{j\theta_m}}{R_{m-n} + jX_{m-n}}\right|$$

Equation 3.5

N-1 contingency analysis

Contingency Analysis of a power system is a major activity in power system planning and operation. Any line, transformer our generator outage could provoke risky over loads in other branches and/or sudden system voltage rise or drop [9]. Therefore, N-1 contingency analysis is a basic and fundamental tool to check operation security and anticipate potential risks.

In practice, what we do is to study the system response to any element outage: branch, transformer or generator. Two main kinds of violations may be caused by these contingencies: low voltage violations and line current/power flow/temperature limit violation. In fact, this last is the one that most concerns us here.

At this point thermal rating has an important contribution as it permits security operators having a deeper and more realistic idea of line thermal limits and margins. That is the reason why this work is focused on incorporating the HBE is power system security analysis (Section 4), and apart from the base case load flow, all the N-1 contingencies will be individually analyzed.

At this document we have not specially focus on Remedial Action Schemes (RAS) that are the different steps our types of remedial action in order to make the system recover and go back to its normal operation once a violation due to a contingency has taken place. However, only one of them, that is generation re-dispatch, has been automatically implemented for generators outage and is explained in detail at section 0.

So basically, what it has been done to run the N-1 contingency analysis using MATPOWER is fixing the status of it generator and branches (transformers are here included according to MATPOWER test cases language) as 0 (out of order or simply off) at their corresponding contingency. In addition, for generator contingencies, the already mentioned re-dispatching or response to power imbalance was applied.

Thus, the total number of contingencies, at this document called N, is equal to: number of network generators + number of network branches (transformers are here included as well).

Model of generators' response to power imbalances

MATPOWER usual response to power imbalances when running the normal power flow is dispatching by default the new mismatch to the generator located at the slack bus. This might became an important problem if the mismatch becomes quite big as this generator might no be able any longer to compensate the power imbalance by its own while others generators with capacity enough are underused.

A quite simple model with the only objective of dispatching generation power more efficiently has been designed and is explained in the following lines.

First of all, we will start with power imbalances due to generators N-1 contingency, which means how do the rest of generators *i* dispatch the generator's load that is supposed to be out of order, here assigned with letter *n*.

For better understanding, we present a short description of each variable:

- *i*: any generator available that can participate on the dispatch (usually all generators except generator *n*).
- *n*: outage generator at that specific moment t; the one that we set out of order to check the N-1 contingency reliability.
- $R_{tot}^{u}\ (t)$: Total upward reserve provided by all generators available at time step t (when generator *n* outage takes place).
- $R_i^u\ (t)(t)$: Individual upward reserve provided by generator *i* at time step t.
- $P_{\max i}(t)$: Maximum power limitation of generator *i* at time step t.
- $P_{g\ i}(t)$: Power load generation of generator *i* at time step t.
- $P_{g\ n}(t)$: Power load generation of generator *n* at time step t. It must be zero by definition.
- $\Delta P_{tot\ g}(t)$: Mismatch or total increment of power load generation due to generator *n* outage at time step t.
- $\Delta P_{g\ i}(t)$: Individual increment of power generation of generator *i* due to generator *n* outage at time step t.

We calculate the total available reserve. To do so, we solve the summation without taking into account generator n.

$$R_{tot}^{u}\ (t) = \sum_{i \neq n} R_i^u\ (t) = \sum_{i \neq n} P_{\max i}(t) - P_{g\ i}(t)$$

Equation 3.6. Total upward reserve for one generator n outage

The total increment of power load generation must be equal to the power that generator n was providing before the outage, so at time step t-1.

$$P_{g\ n}(t) = 0 \rightarrow \Delta P_{tot\ g}(t) = P_{g\ n}(t-1)$$

Equation 3.7. Mismatch or increment of power load generation due to one generator n outage

Now it is possible to calculate the individual increment of each *i* generator. The individual dispatch of power generation is directly proportional to the individual reserve contribution of each *i* generator.

$$\Delta P_{g\ i}(t) = \left(P_{\max i}(t) - P_{g\ i}(t)\right) \cdot \frac{\Delta P_{tot\ g}(t)}{R_{tot}^{u}\ (t)}$$

Equation 3.8. Individual power load increment due to one generator n outage

Finally, just adding the increment with the generation load right before the outage of *n* we arrive to the specific power load of generator *i* at time step t:

$$P_{g\ i}(t) = P_{g\ i}(t-1) + \Delta P_{g\ i}(t)$$

Equation 3.9. Individual power load due to one generator n outage

3.1.1 Incorporation of a wind farm power plant to the system

Nowadays, with the important rise of renewable energies and specially wind power energy, thanks to the potential correlation between wind farm output and dynamic line rating capacity of nearby overhead lines, dynamic line rating can be highly exploited.

In addition, many political incentives promote wind farms and there is a feed-in priority for electricity from renewable sources. Thus, other power plants, such as thermal, are usually subject to dispatch in their first place. Several studies have discussed and presented the potential of improved wind integration by dynamic thermal rating of overhead lines [6] and how does reserve requirement can be enhanced through probabilistic line rating [5], for example.

At this point, we have substituted one of the generators of the network by a wind farm. As feed-in priority policies work with renewable sources, here we have also given the highest priority to exploit wind energy. Therefore, this may provoke power imbalances in the network due to stochastic wind that other generators should compensate.

At Appendix 1 is explained the output power behavior of the wind farm. Also it might be useful to have a look to section 0 too, where weather uncertainties as like as scenario construction methodology is developed as wind farm weather conditions must be exactly the same or at least very similar to nearby overhead lines.

In comparison with what it was presented at the beginning of section 0, here the mismatch our increment of generation of power load generation, $\Delta P_{tot\ g}(t)$, might be positive our negative, always depending on wind farm output and small demand variations (look at section 0, demand uncertainty). To avoid confusions, we show again the list of all variables implicated, it looks alike the one before but there are some differences to highlight. In fact, *wf* will be the wind farm "generator" but it should be reminded that this dispatch process is done for the base case, which means that after that the N-1 contingency analysis must be run too and for that the previous procedure is still perfectly valid and implemented actually.

- *i*: any generator available that can participate on the dispatch (usually all generators except the wind farm *wf*).
- *wf*: the wind farm "generator".
- $R_{tot}^{u}\ (t)\ (t)$: Total upward reserve provided by all generators available at time step t, except from *wf*, which is considered to have no reserve.
- $R_{i}^{u}\ (t)$: Individual upward reserve provided by generator *i* at time step t.
- $R_{tot}^{d}\ (t)\ (t)$: Total downward reserve or power margin (subtraction between actual power generation an minimum power) provided by all generators available at time step t, except from *wf*, which is considered to have no margin.
- $R_{i}^{d}\ (t)(t)$: Individual downward reserve or margin of generator *i* at time step t.
- $P_{\max i}(t)$: Maximum power limitation of generator *i* at time step t.
- $P_{g\ i}(t)$: Power load generation of generator *i* at time step t.

- $P_{g\,wf}(t)$: Power load generation of *wf* at time step t.
- $\Delta P_{tot\,g}(t)$: Mismatch or total increment of power load generation (positive or negative) due to *wf* power output variation at time step t.
- $\Delta P_{g\,i}(t)$: Individual increment of power generation of generator *i* due to *wf* power output variation at time step t.

For this case, the mismatch or total increment of power load generation must be equal to the subtraction between the total demand and the total generation (including the wind farm of course) at time step t.

$$\Delta P_{tot\,g}(t) = P_{tot\,d}(t) - P_{tot\,g}(t)$$

Equation 3.10. Mismatch of power load generation due to wind farm and demand variations

As it was said before, the mismatch may not be always positive this time. The power unbalance should be solved differently in case of positive our negative mismatch indeed.

If it becomes to be positive (upward response of generating *i* units), once again, we calculate the total upward reserve. To do so, we solve the summation without taking into account the wind farm *wf*.

$$R_{tot}^{u}(t) = \sum_{i \neq wf} R_i^u(t) = \sum_{i \neq wf} P_{\max i}(t) - P_{g\,i}(t)$$

Equation 3.11. Total upward reserve of all generators except the wind farm

From there, it is easy to obtain the individual increment of each *i* generator. As like as before, the individual dispatch of power generation is directly proportional to the individual reserve contribution of each *i* generator.

$$\Delta P_{tot\,g}(t) > 0 \rightarrow \Delta P_{g\,i}(t) = \left(P_{\max i}(t) - P_{g\,i}(t)\right) \cdot \frac{\Delta P_{tot\,g}(t)}{R_{tot}^{u}(t)}$$

Equation 3.12. Individual power load increment due to wind farm and demand variations

However, when we deal with a negative mismatch, which implies a downward power response of all *i* units. Instead of the upward reserve, the total downward reserve and individual margin of each generator i should be considered this time:

$$R_{tot}^{d}\ (t) = \sum_{i \neq wf} R_{i}^{d}\ (t) = \sum_{i \neq wf} P_{g\,i}(t) - P_{min\,i}(t)$$

Equation 3.13. Total margin all generators except the wind farm

The individual power downward is directly proportional to the individual margin contribution of each *i* generator.

$$\Delta P_{tot\ g}(t) < 0 \rightarrow \Delta P_{g\ i}(t) = \left(P_{g\,i}(t) - P_{min\,i}(t)\right) \cdot \frac{\Delta P_{tot\ g}(t)}{R_{tot}^{d}(t)}$$

Equation 3.14. Individual power load fall due to wind farm and demand variations

Anyway, either for any of both cases, the new power load of each generator *i*:

$$P_{g\ i}(t) = P_{g\ i}(t-1) + \Delta P_{g\ i}(t)$$

Equation 3.15. Individual power load due to wind farm and demand variations

4 Incorporation of HBE in power system security analysis

Arrived at this point, we will describe the procedure implemented to solve the heat balance equation. Despite having a very powerful tool, as it is MATLAB, it was not evident especially at the beginning which function and method would be the most efficient. It should not be forgotten that the problem is not just simply to solve the complex nonlinear equation but to minimize as much as possible the computation time. At first, when analyzing *deterministic* cases, which is to say that uncertainties are not taken into account, we might not care so much about that. However, as it will be explained at section 5, to deal with uncertainties sampling an N number of scenarios is absolutely necessary. Therefore the total number of calculations increases by a factor of N times. For example, if we select IEEE test case 30, that counts with 30 buses, 41 lines and 6 generators. At a specific time moment, running the power load flow for the base case and the N-1 contingencies would mean solving the heat balance equation in total: 1 (base case)+ 6 (number of generators)+ 41 (number of lines) = 48. On the other hand, willing to obtain smooth results and consider different uncertainties, if we fix a reasonable number of sampled scenarios, lets say 60, the total calculations raise to: 48 × 60 = 2,880. Up to here, we have just considered one specific moment, but in case that we want to obtain one day hourly rates: 48 × 60 × 24 = 69,120. So we can conclude that this is an important matter to care about.

As it was briefly presented at section 2,the nonlinear dependence between the internal variables of the HBE makes complex the resolution of this, especially those of the convection term which equations can be found below.

First, we remind the value of T_f as the average between the ambient and the line temperature:

$$T_f = \frac{T_a + T_l}{2}$$

Equation 4.1. Film temperature [°C]

$$\mu_f = \frac{1.458 \cdot 10^{-6} \cdot (T_f + 273)^{1.5}}{T_f + 383.4}$$

Equation 4.2. Dynamic viscosity of air [Pa·s]

$$\rho_f = \frac{1.293 - 1.525 \cdot 10^{-4} \cdot H + 6.379 \cdot 10^{-9} \cdot H^2}{1 + 0.00367 \cdot T_f}$$

Equation 4.3. Air density [kg/m^3]

$$k_f = 2.424 \cdot 10^{-2} + 7.477 \cdot 10^{-5} \cdot T_f - 4.407 \cdot 10^{-9} \cdot T_f^2$$

Equation 4.4. Thermal conductivity of air [W/(m·°C)]

Before explaining how the HBE could be solved, here we reproduce main steps that will be always carried on in order, there might be some internal variations but mainly we will proceed like this always:

1) Upload to MATLAB the network case (IEEE test cases or any other real one converted to MATLAB code).
2) Upload/fix: the weather parameters of each line (ambient temperature, wind speed, solar heat) and line conductors specifications (T_{ref}, R_{ref}, D).*
3) Run the load flow through MATPOWER.
4) Storage of all line power flows.
5) Calculation (with the difference between the two connected buses' voltages divided by the line impedance) and storage of line currents in per units and after that in real values.
6) Resolution of the non-linear heat balance equation in order to calculate the temperature of each line.
7) Verification of non-maximum temperature exceeded at any line or case and results analysis.

Steps 3), 4), 5), 6) and 7) will be done N times (base+ N-1 contingencies) for the deterministic evaluations, while N × Number of scenarios times for the probabilistic one.

*When dealing with the probabilistic evaluation, additional information may be required such as: number of scenarios and probabilistic parameters of weather and demand distributions. Look at section 0 for more detailed information.

Next sections 0 and 0 are entirely devoted to describe two possible methods for step 6).

Iterative process

Initially, the high dis-linearities at the HBE, especially provided by the part of the convection parameters: ρ_f, μ_f and k_f, were faced though an iterative process where instead of coupling their exact equations, for each iteration step concrete values of ρ_f, μ_f and k_f were taken from the following table that can be found it at IEEE 738 [15].

Temperature T_{film}	Dynamic discosity μ_f	Air density ρ_f (kg/m³)				Thermal conductivity of air k_f
°C	(Pa·s)	0 m	1000 m	2000 m	4000 m	W/(m·°C)
0	0.0000172	1.293	1.147	1.014	0.785	0.0242
5	0.0000174	1.270	1.126	0.995	0.771	0.0246
10	0.0000176	1.247	1.106	0.978	0.757	0.0250
15	0.0000179	1.226	1.087	0.961	0.744	0.0254
20	0.0000181	1.205	1.068	0.944	0.731	0.0257
25	0.0000184	1.184	1.051	0.928	0.719	0.0261
30	0.0000186	1.165	1.033	0.913	0.707	0.0265
35	0.0000188	1.146	1.016	0.898	0.696	0.0269
40	0.0000191	1.127	1.000	0.884	0.685	0.0272
45	0.0000193	1.110	0.984	0.870	0.674	0.0276
50	0.0000195	1.093	0.969	0.856	0.663	0.0280
55	0.0000198	1.076	0.954	0.843	0.653	0.0283
60	0.0000200	1.060	0.940	0.831	0.643	0.0287
65	0.0000202	1.044	0.926	0.818	0.634	0.0291
70	0.0000204	1.029	0.912	0.806	0.625	0.0295
75	0.0000207	1.014	0.899	0.795	0.616	0.0298
80	0.0000209	1.000	0.887	0.783	0.607	0.0302
85	0.0000211	0.986	0.874	0.773	0.598	0.0306
90	0.0000213	0.972	0.862	0.762	0.590	0.0309
95	0.0000215	0.959	0.850	0.752	0.582	0.0313
100	0.0000217	0.946	0.839	0.741	0.574	0.0317

Table 4.1. Viscosity, density and thermal conductivity of air.

Below, it is shown the iterative process that is explained step-by-step right after:

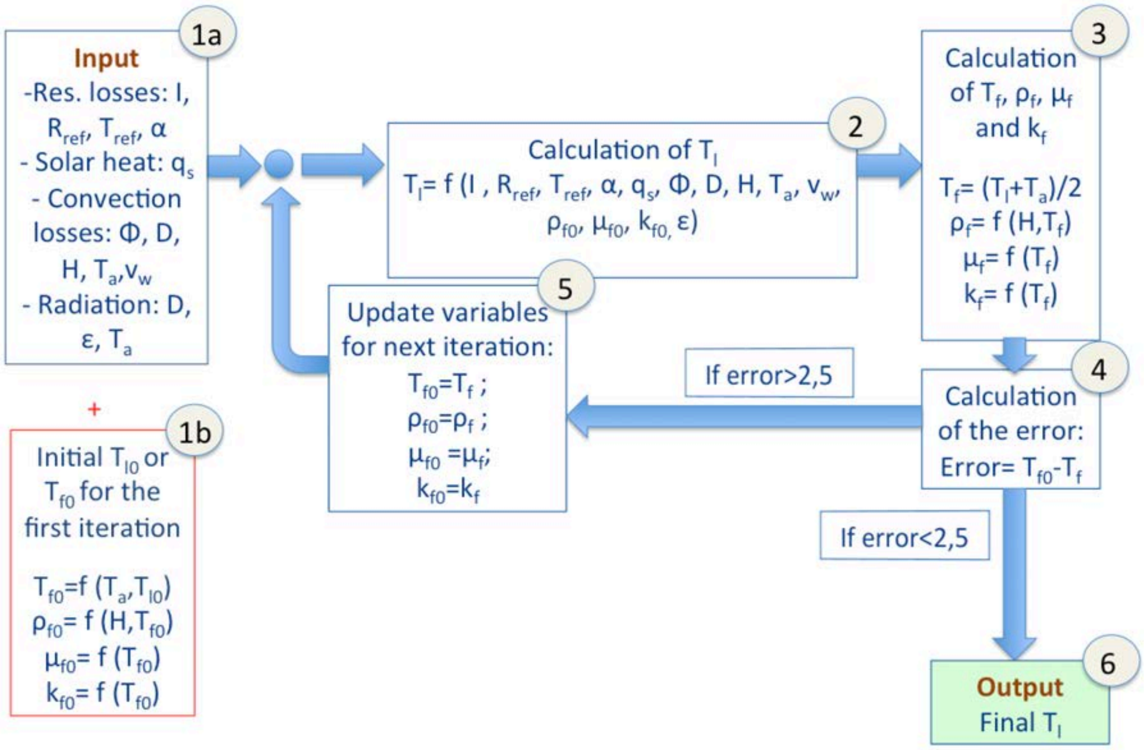


Figure 4.1. Iterative process HBE

- Step (1a) collects previous steps 2), 3) and 4) where weather conditions and conductors specifications were fixed and, in addition, the power load flow was run in order to obtain from it the currents of each line.

- However, an additional step here called (1b) is necessary to initialize the iteration process. This one will be done just once per each line temperature calculation and always at the beginning. As it was already mentioned, for this procedure ρ_f, μ_f and k_f are not going to be exactly calculated from Equation 4.2, Equation 4.3 and Equation 4.4 but directly obtained from Table 4.1. Therefore, it will be supposed and initial T_{l0} or T_{f0} (one can be directly deduced from the other as T_a is known) and then just going to Table 4.1. it can be taken the corresponding ρ_{f0}, μ_{f0} and k_{f0} as we also know H (1, 1000, 2000 or 4000m).

- Of course, this T_{l0} and T_{f0} may differ with T_l that is calculated at step (2) thanks to the input that receives from (1a) and (1b) for the first iteration and afterwards from (5)

- Then, at step (3) through the new T_l we are able to obtain the new values of T_f, ρ_f, μ_f and k_f.

- As it was already said, the new parameters ρ_f, μ_f and k_f may differ from the initial ρ_{f0}, μ_{f0} and k_{f0} as like as T_{f0}/T_{l0} and T_f/T_l, that is why at step (4), the error between both, T_{f0} and T_f, is checked.

- As the T_f variation between one row and another is of 5°C, is has been fixed that the maximum acceptable error must be 2.5; otherwise the iterative process will be repeated but before it starts again it is necessary to update T_{f0}, ρ_{f0}, μ_{f0} and k_{f0} but from the previous T_f, ρ_{f0}, μ_{f0} and k_{f0} in order to get each time closer to the right solution. Thus, if the error is greater than 2.5 we move on to step (5) and from there back to (2) and so on.

- Finally, at some point after the needed iterations (quite few our close to one if T_{f0} is well-selected), the error becomes lower than 2.5 and T_l is accepted as final solution and stored.

It can be easily guessed; a good choice of T_{f0} is determinant to speed up the iterative process. Despite of having a quite good performance with this method for a good number of lines, while dealing with uncertainties and probabilistic methods, the number of calculations increments enormously and that is the reason why next section 0 has been developed as the iterative process was not fast enough.

Exploiting natural convection term correlation

At this point, we will explain an important fact that will permit us to do a good approximation and after that an important computation time saving improvement.

To do so, an auxiliary variable, T_c, is defined. Actually, T_c is the interval where the future T_l variable must be. Therefore, its minimum is T_a and for the maximum we will fix a high enough value to be sure that T_l will be always inside the interval, for example 150°C.

Now, for each value of T_c we calculate the corresponding convection parameters ρ_{f0}, μ_{f0} and k_{f0} using the appropriate equations instead of Table 4.1 as it was done at section 0. Once we have the convection parameters for each T_c, with Equation 2.10 and Equation 2.11 it is easily obtained q_{c1} and q_{c2}.

Though, up to here we can not know what is the exact value of T_l, we can only assure that it must belong to the interval T_c. But, when we represent the evolution of the convective term along T_c we discover and interesting fact to take advantage of. There is a well approximate linear correlation between T_l and q_c as it can be seen at Figure 4.2.

On the x-axis, we find the variable T_{ca} that is not but the subtraction between T_a and T_c.

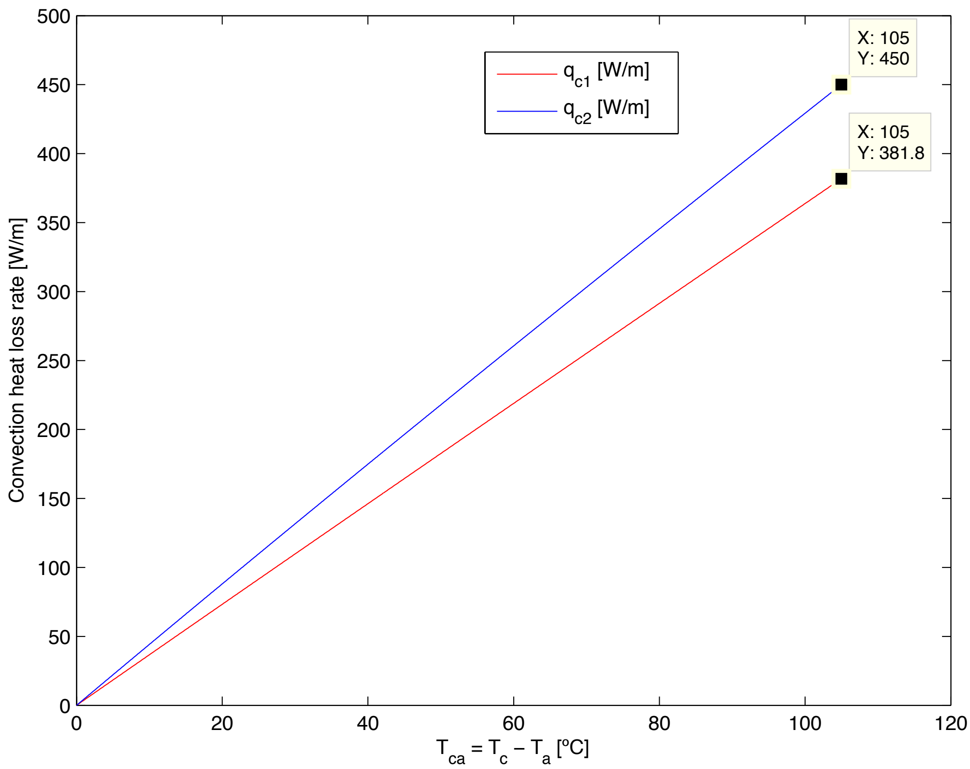


Figure 4.2. Linear evolution of q_{c1} and q_{c2} in function of the line temperature.

Once again, we insist that this is a great advantage as since now, instead of iterating each time, just calculating the maximum slope between q_{c1} (q_{cm1}) and q_{c2} (q_{cm2}), it is possible to substitute directly this term in the HBE as the slope keeps constant for any value of T_l.

Just as it is shown below and it can be deduced from equations Equation 2.10 and Equation 2.11 at section 0:

$$q_{c1} = \underbrace{\left[1.01 + 0.0372 \cdot \left(\frac{D \cdot \rho_f \cdot v_w}{\mu_f}\right)^{0.52}\right] \cdot k_f \cdot K_{angle}}_{} \cdot (T_l - T_a)$$

$$q_{c1} = q_{cm1} \cdot (T_l - T_a) = q_{cm1} \cdot T_{ca}$$

Equation 4.5. q_{c1} linear equation

$$q_{c2} = \underbrace{\left[0.0119 \cdot \left(\frac{D \cdot \rho_f \cdot v_w}{\mu_f}\right)^{0.6}\right] \cdot k_f \cdot K_{angle}}_{} \cdot (T_l - T_a)$$

$$q_{c2} = q_{cm2} \cdot (T_l - T_a) = q_{cm2} \cdot T_{ca}$$

Equation 4.6. q_{c2} linear equation

An also, $q_{cmax} = \max\left(q_{cm1}, q_{cm2}\right)$. Therefore the new HBE results to be for forced convection quite simpler:

$$0 = I^2 \cdot R_{ref} \cdot \left(1 + \alpha \cdot \left(T_l - T_{ref}\right)\right) + q_s - q_{cmax} \cdot (T_l - T_a) - 0.0178 \cdot D \cdot \varepsilon \cdot \left[\left(\frac{T_l + 273}{100}\right)^4 - \left(\frac{T_a + 273}{100}\right)^4\right]$$

Equation 4.7. Simplified HBE with forced convection

Deterministic evaluation for a given weather parameters

Once the HBE resolution problem has been satisfactorily solved, we are able to implement the previous procedures (in fact since now we will always exploit forced convection term linear behavior as it saves quite a lot of computation time) in order to calculate the lines temperatures that is indeed our main goal.

Section 5 is entirely dedicated to uncertainties modeling and how to deal with them. However at this point we will just focus on deterministic evaluation, which means that it will be supposed that all the inputs are determined or concrete (number of scenarios equal to one). For example, all weather conditions are supposed to be exactly known at a specific time moment and, therefore, this permits us to obtain a specific and deterministic solution for each line.

Moreover, as it was said at section 0, security analysis is one of the principal applications of this work. Thus, apart from the normal case, the N-1 contingencies analysis will be always done. That is the reason why this tool has been called Deterministic N_Contigencies, its input-output diagram is right below.

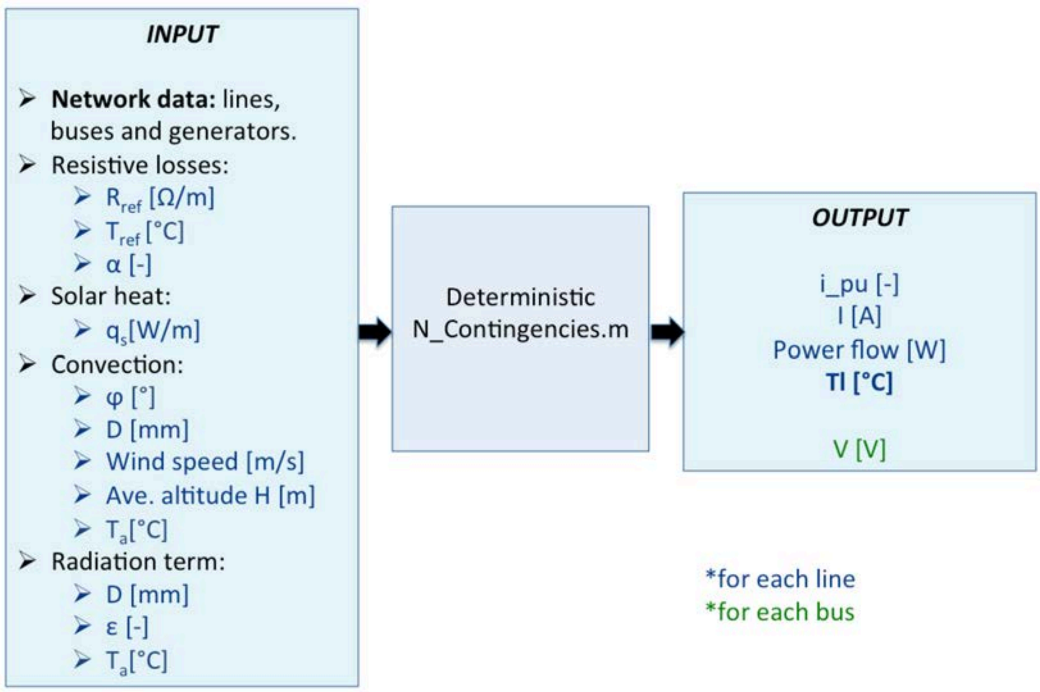


Figure 4.3. Deterministic N_Contingencies.m

Basically, N_Contingencies.m is running the 7 steps explained at the beginning of section 0 for just one scenario (deterministic).

5 Modeling of uncertainties

Deterministic evaluation may be very useful to implement for real-time provided data as that way the values introduced (inputs) have been measured and thus we are quite certain that they are realistic. However, while forecasting, deterministic evaluation loses its own sense due to the wide range of uncertainties that are not considered by that evaluation.

We devote this part to explain the difference sources of uncertainties that we face up to and how there are modeled.

It is possible to divide these sources into two categories:

- Weather uncertainties
- Node demand uncertainties

Weather uncertainties

Despite weather conditions may have a certain effect on line parameters such as impedance because of humidity and ambient temperature [4], here we will only consider the effect that weather conditions have on the HBE. Therefore, we try to find a good model for these uncertainties that are in fact the input of our system.

In our case, the main sources that we can consider are: ambient temperature T_a, solar heat q_s, wind speed v_w and wind direction ϕ. Nevertheless, the last one, ϕ, quite often is easier to predict with certitude and also a conservative value can be enough to obtain robust results. Actually, at this work only the first three uncertainties were modeled with the following distributions:

- Ambient temperature: $T_a \approx N(\mu_{Ta},\sigma_{Ta})$

- Wind speed: $V_w \approx W(k_{Vw}, \lambda_{Vw})$

- Solar heat: $q_s \approx N(\mu_{qs},\sigma_{qs})$

Then, for the ambient temperature we use a normal distribution as like as for the solar heat. On the other hand, wind speed behavior is better designed with a Weibull distribution. On the next table we show some of the most common values parameters that we might find:

Ambient temperature [°C]		***Wind speed [m/s]***		***Solar heat [W/m]***	
Normal distribution		Weibull distribution		Normal distribution	
μ_{Ta}	σ_{Ta}	k_{Vw}	λ_{Vw}	μ_{qs}	σ_{qs}
5-28	2-5	2-7.5	1.5-3.5	3-16	1.5-3

Table 5.1. Common weather probabilistic parameters

Demand uncertainty

Apart from the already mentioned weather parameters, consumers demand may be also subjected to uncertainty. Usually, the power factor remains constant despite of active and reactive power variation. Therefore the active and the reactive power, will be modeled as normal distributions with a small standard distribution (1% of the average or expected value) and taking into account that power factor must keep constant at each bus.

- Demand: $P_d \approx N(\mu_{Pd}, \sigma_{Pd})$

Where: $\sigma_{Pd} \approx 0.01 \cdot \mu_{Pd}$

Scenario construction methodology: Stochastic evaluation

In order to assemble all this uncertainties, stochastic sampling will be necessary. At section 5.1.1, Latin Hypercube Sampling (LHS) method, which is the main statistical method for generating samples that has been used, is explained.

In contrast with the deterministic evaluation at section 0, now we will generate a number of samples (or scenarios) that belong to the probabilistic distributions so that we will also represent the solution in probabilistic terms. Thus, an additional input here defined as: ***Number of scenarios,*** is to be introduced.

Inside the input block we can see in black the new variables or those that somehow have changed in comparison with Deterministic N_Contingencies.m

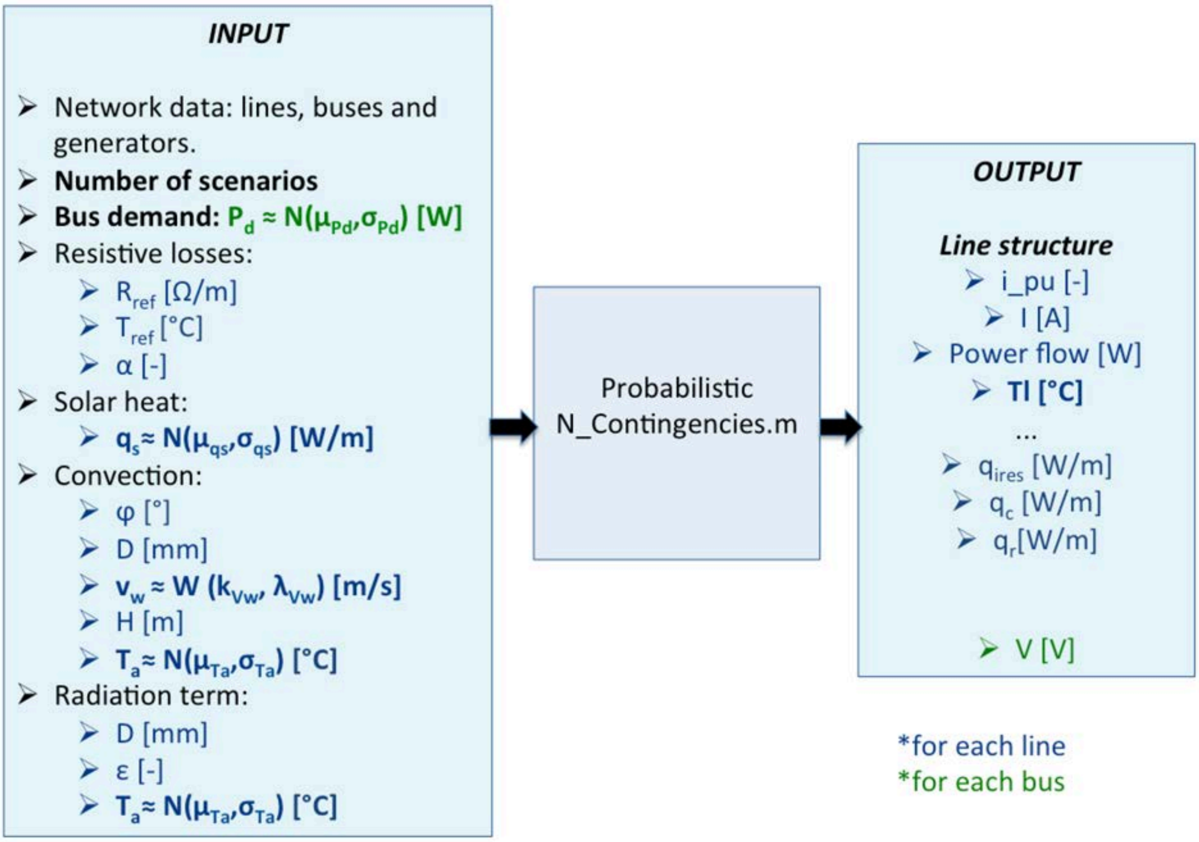


Figure 5.1. Probabilistic N_Contingencies.m

Storing all the results in an efficient way was one of the main challenges. We remind the reader that the number of computations has raised one dimension. Before we were doing all the calculations (i_pu, I, T_l...) per line and per contingency. Now, it should be done per line per contingency and per scenario. To help visualization, at Appendix 1, an overview of outputs storage is given.

5.1.1 Latin Hypercube Sampling (LHC) versus Monte Carlo Sampling (MCS)

Sampling is not an evident matter. Lots of sampling methods have been developed and MATLAB provides also several.

Monte Carlo sampling uses random or pseudo-random numbers to sample from a probability distribution. However, the main problem that we could find to this is to know what is the stopping criterion, which means, how many random samples do we need to obtain a reasonable representation of the input distribution.

We do not look here to insist too much on statistical methods, but through the Central Limit Theorem (CLT) of statistics, it is possible to calculate the number of necessary samples applying the concept of the standard error of the mean (SEM) and fixing the desired confidence interval. [10]

On the other hand, Latin Hypercube Sampling (LHC) method is able to match the input distribution very closely with just few samples. This is achieved thanks to division of the cumulative curve into equal intervals on the cumulative probability scale and, once the intervals are defined, it takes random values within the interval. Thus, Latin Hypercube method is more efficient and makes simulation converge much faster than Monte Carlo sampling; this is mainly why LHS method was the chosen one for sampling.

Figure 5.2 shows the different ambient temperature distribution obtained through random sampling and Latin Hypercube sampling for 50 samples. On the other hand, Figure 5.3 shows exactly the same sampled distribution but for 10,000 samples.

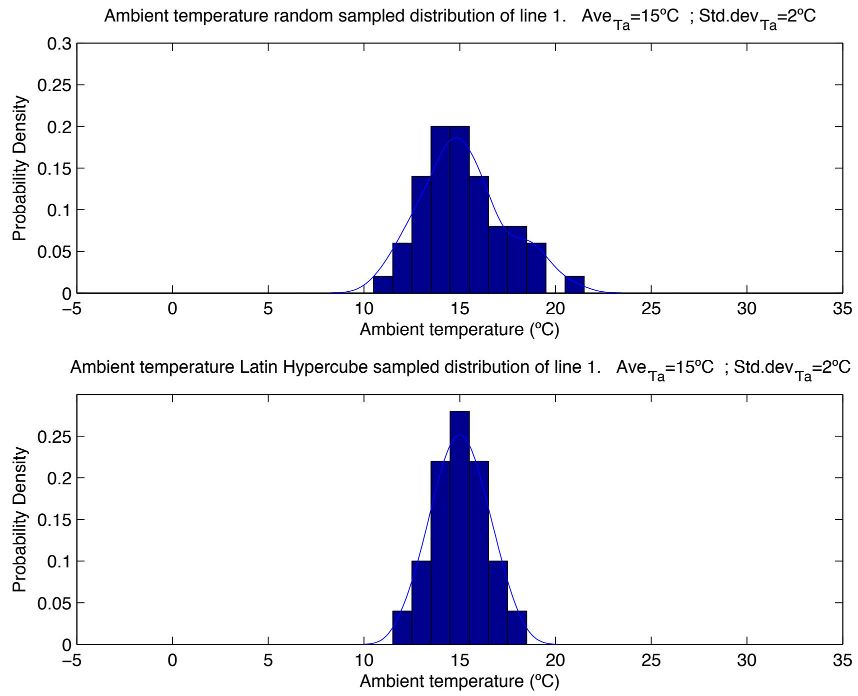


Figure 5.2. Comparison of random sampling with LHS for 50 samples

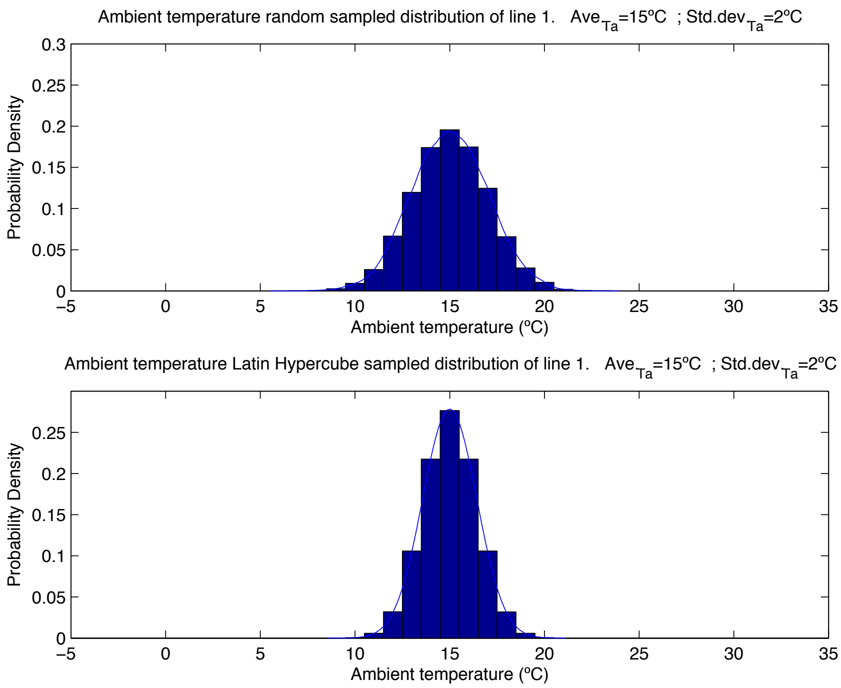


Figure 5.3. Comparison of random sampling with LHS for 10'000 samples.

6 Simulation results

Test case studies: input variables.

As it was shown at the beginning of section 0 and also at Figure 5.1, the MATLAB program Probabilistic N_Contingencies.m samples with the probabilistic parameters (average and standard deviation if it is a normal distribution and scale and shape parameters if it is a Weibull one) and introduces as an input the distribution of the uncertain variables (demand and weather variables).

All the results presented at this section have been obtained from a simulation fixed with 500 scenarios, in order to obtain smooth distribution results. Since through LHS the input distributions are already smooth and accurate with a quite small number of samples as it was explained in the previous sections.

Below, the input distribution of the ambient temperature, wind speed and solar heat of line 35 are displayed. These results can also be compared with the vertical red lines that represent the conservative and/or static rating. At the top of each graph data referred to the distribution parameters has been added.

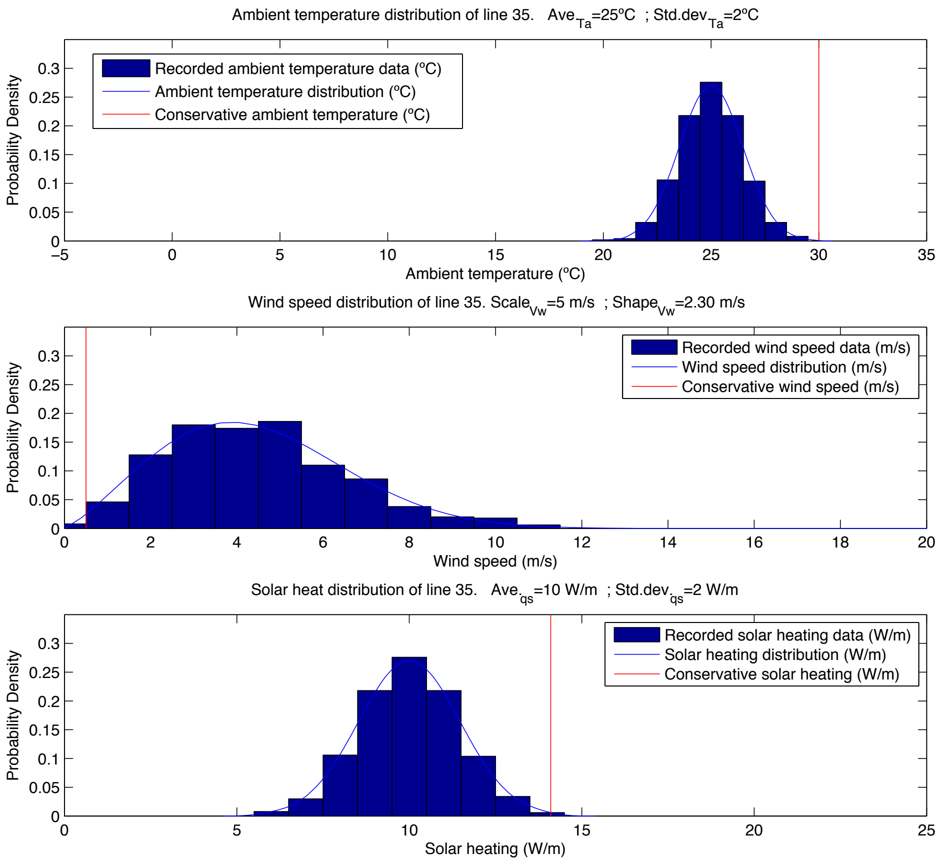


Figure 6.1. Input variables of line 35: T_a, W_d and q_s

At next section we will see for different lines with distinct weather conditions how the output line temperature varies. By the time we observe that for line 35 the weather conditions are not very far from the conservative ones, actually at section 0 we can also see temperature distribution result compared with the static case for these exact weather conditions.

Output variable: line temperature distribution. Comparison between two lines with similar load but different weather conditions.

Right now we have moved on to lines 1 and 24. The input distributions are also shown. It can be appreciated how line 24 weather conditions are quite proximate to the conservative ones. In contrast, line 1 distributions of temperature and wind speed are further. Of course this should have consequences regarding the conductor temperature of both lines. However, another factor should be taken into account before arriving to any conclusion: the load or the current across the lines.

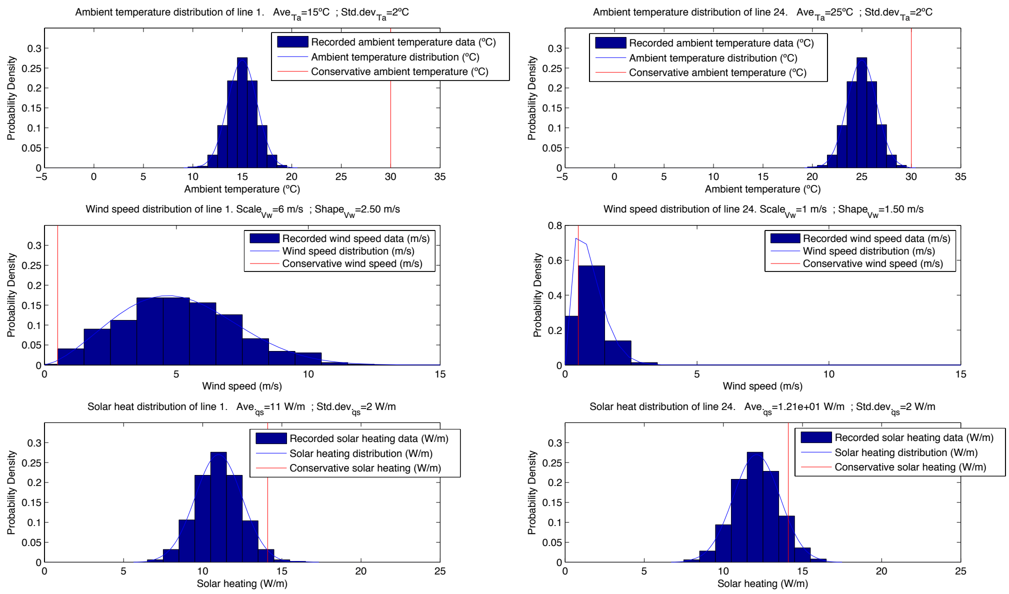


Figure 6.2. Input variables comparison between lines 1 and 24: T_a, W_d and q_s

Actually, at figure Figure 6.3 we see that considering not only the base case but also all the N-1 contingencies, line 1 is much more loaded than line 24. Despite this, the already mentioned weather conditions play an important role and heat up much more line 24, which could exceed at some point the conservative and deterministic temperature.

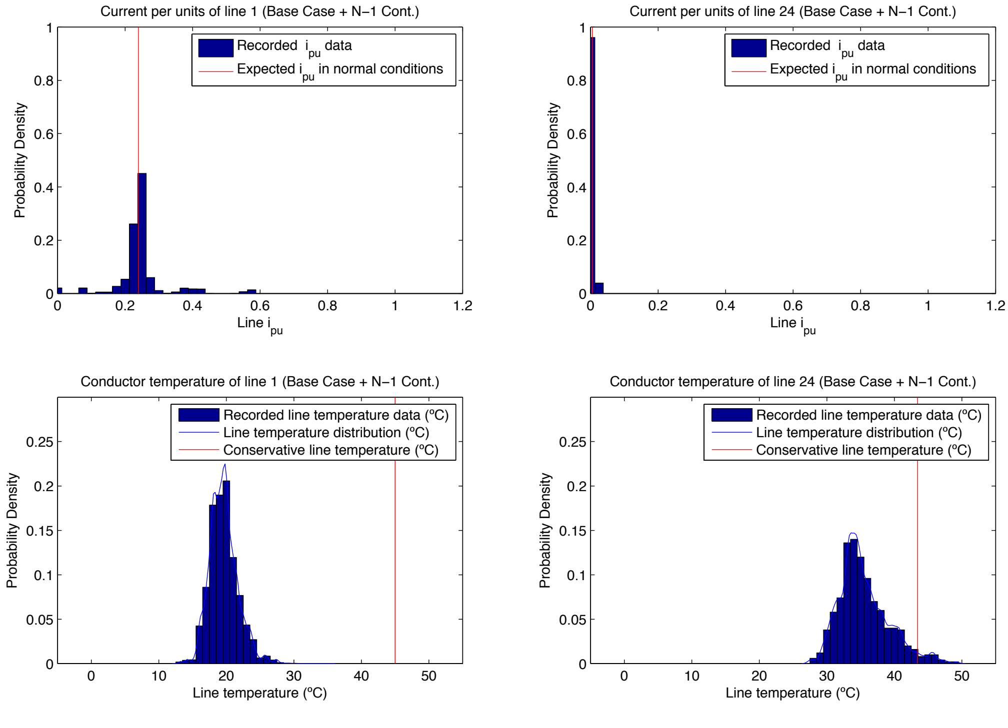


Figure 6.3. Current per unit and conductor temperature comparison between lines 1 and 24

To have a better view, below we compare the CDF of both lines:

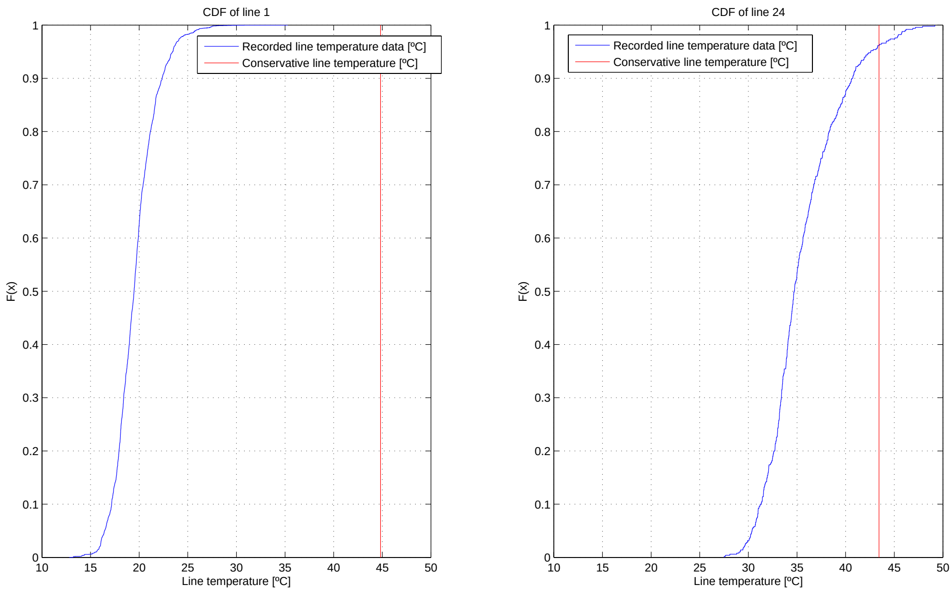


Figure 6.4. CDF line temperature comparison between lines 1 and 24

Comparison of the temperature distribution: All scenarios vs Violating current limits contingency

Going back to line 35, at Figure 6.5 it has been represented the current probability for the base case, when none outages take place, and for all cases (base case and all the rest of contingencies). Therefore, in normal conditions it seems that there might not be any risk of overcurrent. However, it we look carefully the graph below of the current probability for all cases, we can distinguish very small probability but anyway overloads (current higher than 1 p.u.) at the level of 1.2 p.u. For sure, if there are no violations at the base case, this implicates that some contingencies must be checked because current is excessing its limits. In addition, it may be interesting to analyze the expected conductor temperature for that violating current limits contingency (ies).

Checking the results, it has been discovered that contingency 42 (outage of line 34) corresponds to the most extreme case in terms of current rate. This is also shown at Figure 6.6. Apart from the currents, it has been plotted the conductor temperature distribution concerning all cases and only contingency 42. Though, it seems that even with that current growth at that specific contingency, the weather conditions (look Figure 6.1) and quite probably due to the high wind range at that line, the conductor temperature hardly changes between one case and another.

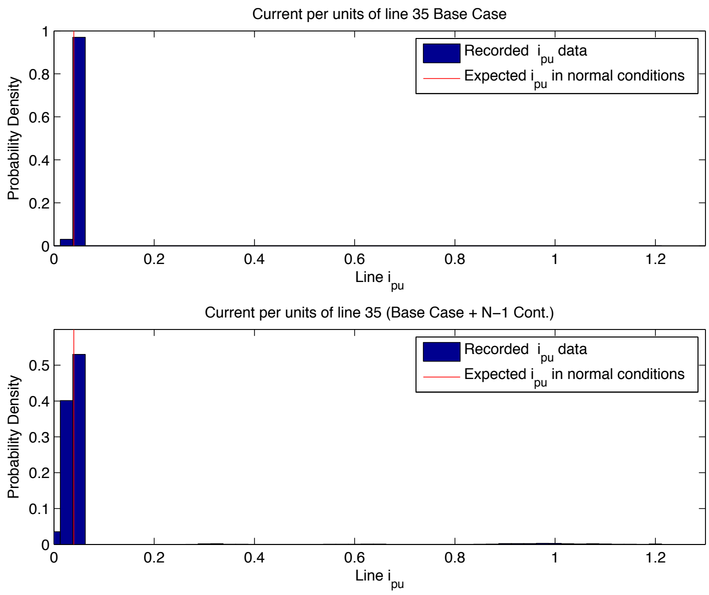


Figure 6.5. Comparison between the base case and all cases (Base Case+ N-1 Contigencies) of the current probability of line 35.

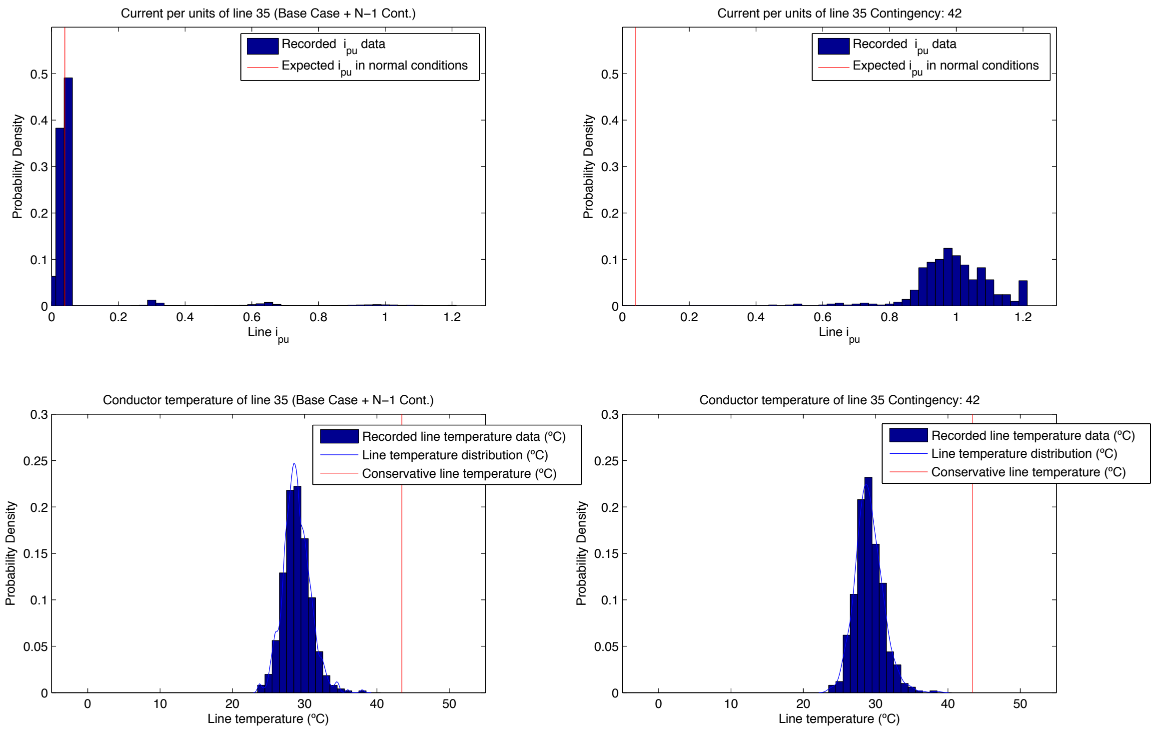


Figure 6.6. Comparison between all cases (Base Case+ N-1 Contigencies) and contingency 42 of the current and conductor temperature probability of line 35.

Line temperature evolution changing weather and load conditions

At this point we found interesting to analyze what is line temperature behavior when we fix constant three of the four modulating variables and we make vary the fourth one.

When taking the constant fixed values, it was tried that they were not too extreme so that this will permit us to see a more natural evolution of the temperature. These values are shown over the table below.

T_a [°C]	V_w [m/s]	q_s [W/m]	i_{pu} [-]
20	1	12	0.5

Table 6.1. Fixed values at figure 6.7

Just to remark the strong effect of the wind speed over the conductor's temperature due to forced convection. Also the current of course, as it was expected to happen. The ambient temperature establishes always the minimum temperature that the conductor may have indeed.

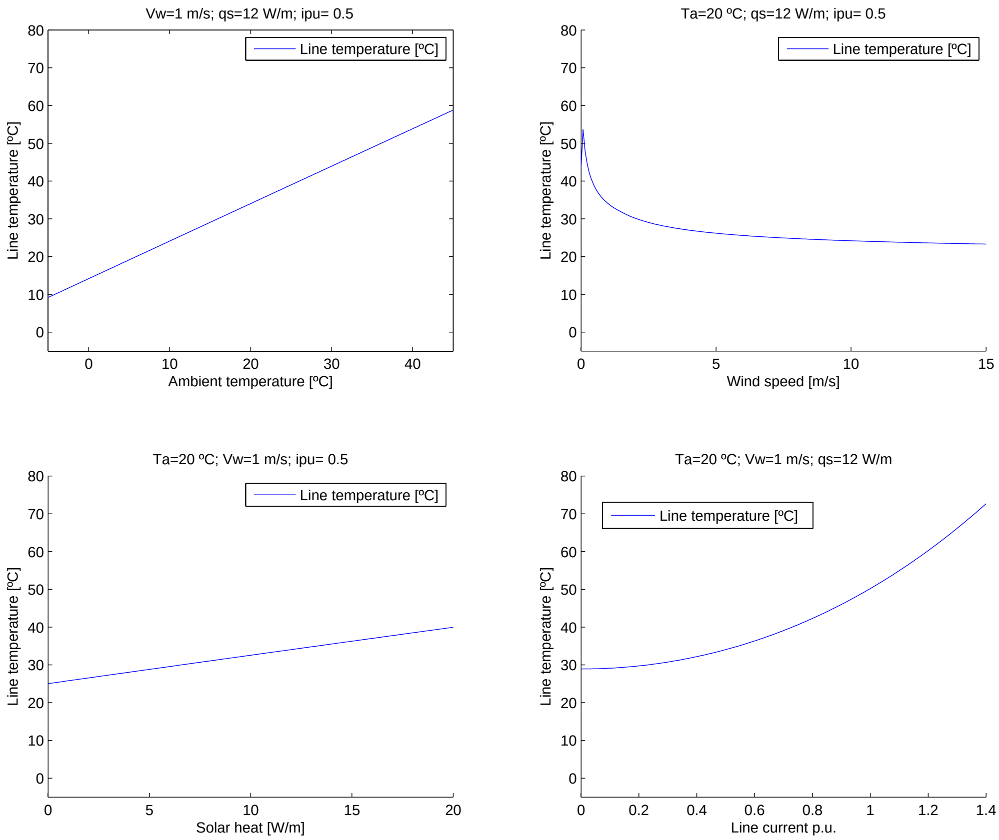


Figure 6.7. Line temperature evolution for different weather and load scenarios

7 Conclusions

At this work we have specially focused on the integration of dynamic thermal rating model of transmission capacity to Security Analysis, however, as many other studies have presented and done deep research on, probabilistic transfer capacity can be very useful also for system planning, contract design and market analysis. [4]

Deterministic evaluation of the HBE has been utilized to compare the static, and usually conservative, thermal rate with forecasted stochastic evaluation. Anyway this may be very useful if real-time provided data of demand and weather conditions are incorporated to the system and thus simulate in real time the conductor's temperature. Of course, for that case, we would not be working on steady state conditions anymore.

Regarding the studied probabilistic evaluation, good weather and load forecast becomes absolutely necessary in order to obtain realistic line temperature forecast. Therefore, and important effort on collecting real data of them would be required because that is on what forecasting should be based.

Even if LHS performance seems to be good when the number of samples is not quite high, a stopping criterion to fix the number of trials required to make sure that the output converges is also recommended.

Finally, as it has been mentioned, a versatile tool was tried to be developed by the way that different network data (previously format adapted to MATLAB) could be uploaded to run the simulation. In fact, all the results shown in this report have been carry out with IEEE case test network of 30 buses, nevertheless, this could have been done without any problem with other IEEE case test or real networks with the beforehand appropriate format adaption.

Bibliography

[1] H. BANAKAR, N.ALGUACIL and F.D.GALIANA. *Electrothermal Coordination Part I: Theory and Implementation Schemes*

[2] H. BANAKAR, N.ALGUACIL and F.D.GALIANA. *Electrothermal Coordination Part II: Case Studies*

[3] MAI RUIKUN, FU LING and XU HAIBO. *Dynamic Line Rating Estimator with Synchronized Phasor Measurement*

[4] I. DOBSON, S. GREENE, R. RAJARAMAN, C.L. DE MARCO, F.L. ALVARADO, M. GLAVIC, J. ZHANG, R. ZIMMERMAN. *Electric Power Transfer Capability: Concepts, Applications, Sensitivity and Uncertainty.*

[5] B. BANERJEE, D. JAYAWEERA and S.M. ISLAM. *Impact of wind forecasting and probabilistic line rating on reserve requirement*

[6] T. RINGELBAND, M. LANGE, M. DIETRICH and H-J HAUBRICH. *Potential of Improved Wind Integration by Dynamic Thermal Rating of Overhead Lines*

[7] M. ZHAO, Z. CHEN and F. BLAABJERG. *Probabilistic capacity of a grid connected wind farm based on optimization method*

[8] MATPOWER 4.1. User's Manual

[9] V.J. MISHRA and M.D. KHARDENVIS. *Contingency Analysis of Power System*

[10] PALISADE: http://kb.palisade.com/index.php?pg=kb.page&id=28

[11] M. KERAMT and R. KELBASE. *Latin Hypercube Sampling Monte Carlo Estimation of Average Quality*. Available: http://www.eecs.berkeley.edu/~alanmi/research/timing/papers/lhsaqi.pdf

[12] INVESTOPEDIA website:
http://www.investopedia.com/terms/m/montecarlosimulation.asp

[13] BU College of Engineering official website:
http://people.bu.edu/dew11/turbineperformance.html

[14] PIKE official website:
http://www.pike.com/EnergySolutions/ThermalRate/TechnicalInformation.aspx

[15] IEEE 738-2006: *Standard for Calculating the Current-Temperature of Bare Overhead Conductors*

Appendix 1: Wind farm output

Looking to analyze the power unbalance of the network caused by wind farms as like as the integration of line temperature rating with them, as it was introduced in section 3.1.1, one of the generators of the network was replaced by a wind farm composed by N variable speed wind turbines.

One of the main characteristics of these turbines is that they operate at the rated power with power regulation during the periods of high wind by the active control of the blade pitch angle or the passive regulation based on aerodynamic stall [7].

Below we show the active power, P_{var}, provided by one variable speed wind turbine in function of the wind speed v_w. Other important conceptual variables are:

- v_i: cut-in wind speed
- v_r: rated wind speed
- v_o: cut-out wind speed
- P_r: rated power of the wind turbine
- a: bias value
- b: gradient value

$$P_{var} = \begin{cases} 0, & 0 \leq v_w < v_i, \\ a + b \cdot v_w^3, & v_i \leq v_w < v_r, \\ P_r & v_r \leq v_w < v_{o,} \\ 0, & v_w > v_o \end{cases}$$

The constant parameters a and b are given by the following formulas:

$$a = \frac{P_r \cdot v_i^3}{v_i^3 - v_r^3} \qquad ; \qquad b = \frac{P_r}{v_r^3 - v_i^3}$$

Next figure shows the output power, P_{var}, evolution for different wind speeds as like as the characteristic wind speed values that define the turbine's behavior.

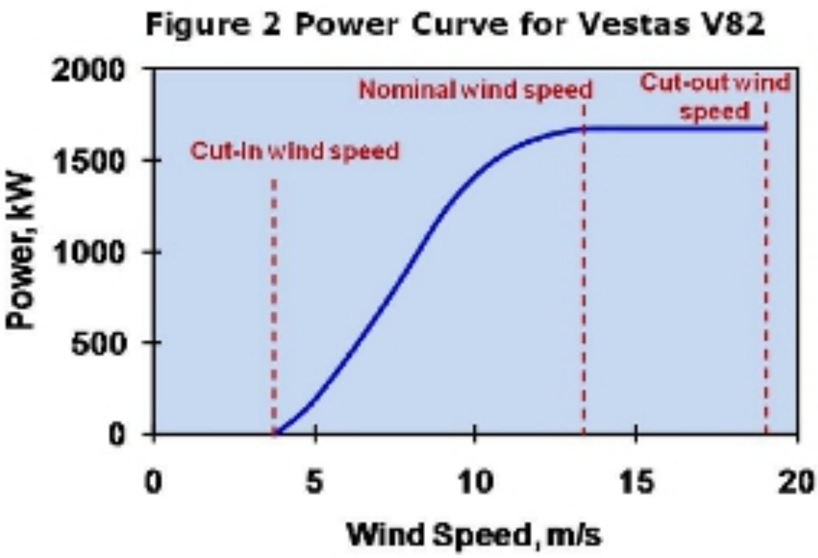


Figure A1. 1. Variable speed wind turbine power curve [13]

To conclude, the total wind power of the wind farm assuming to have N variable speed wind turbines, logically all must be of the same kind and with the same technique characteristics: cut-in wind speed, rated wind speed, cut-out wind speed, etc.

$$P_{farm} = P_{var}(v) \cdot N$$

Appendix 2: ProbabilisticNContingencies output: line structure.

At it was previously introduced in section 0, Probabilistic N_Contingencies.m creates as output a structure here defined as: **line.**

From first view, we can see that each row represents one line: row 1 represents line 1, row 2 represents line 2, etc. And, on the other hand, for each column we have different fields where inputs, auxiliary variables and outputs have been stored. For example, going from left to right we can distinguish the line diameter D [mm], the altitude [m], ambient temperature T_a[°C],(…), temperature of the line T_l [°C], line current in per units and in amperes,(…), q_{cm1}, q_{cm2}, etc.

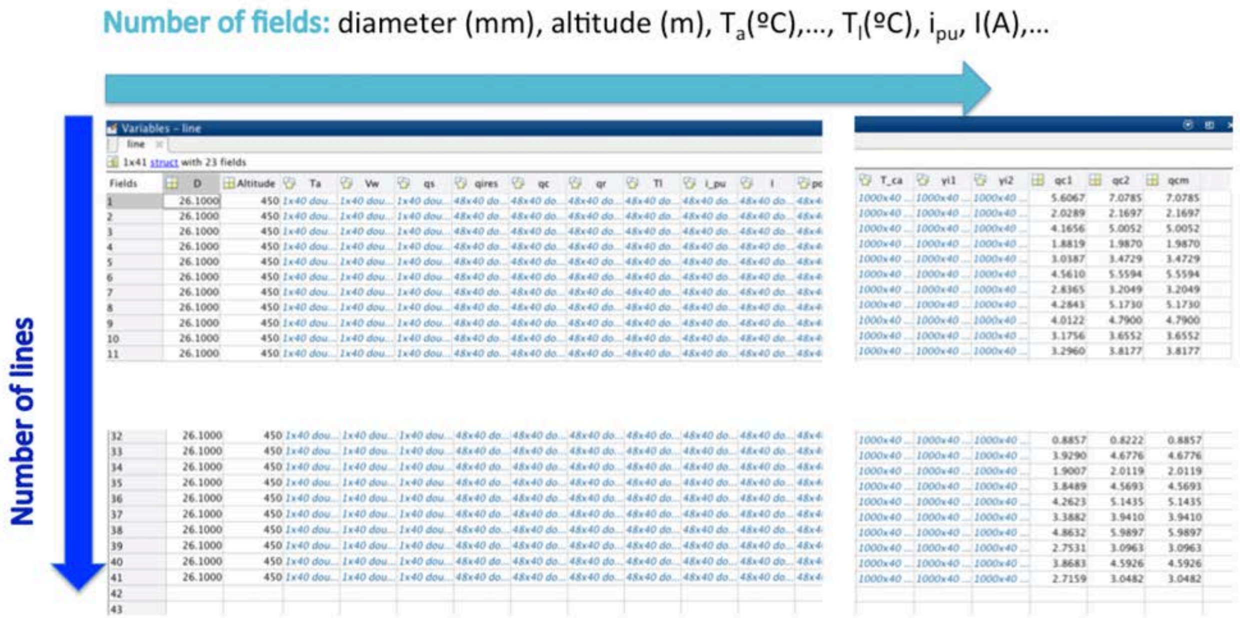


Figure A2. 1 Line structure zoom 1

If we click, for example, to see row 6 (line 6) and field "Ta", we will find inside a row of length 40 (for this case the number of scenarios was 40 then) with a series of numbers that represent the samples that have been given from the ambient temperature normal distribution. We can figure out just observing the results that it might have a mean value close to 25°C and a standard deviation around 3°C.

Figure A2. 2. Line structure zoom 2

However, selecting somewhere else, this time over row 3 (data line 3) and "Tl", we will access then to line temperature results of line 3. Now, instead of a vector, we open a matrix with dimensions: Number of contingencies (rows) × Number of scenarios (columns). Of course, the ambient temperature, at a specific scenario, must be the same for all the contingencies; that is why on Figure A2. 1, line(6).Ta only has one row. On the other hand, at a specific scenario, despite having exactly the same weather conditions (ambient temperature, wind speed and solar heat) and demand load, at each contingency the current may vary at that implies that line temperature should be calculated and may change too.

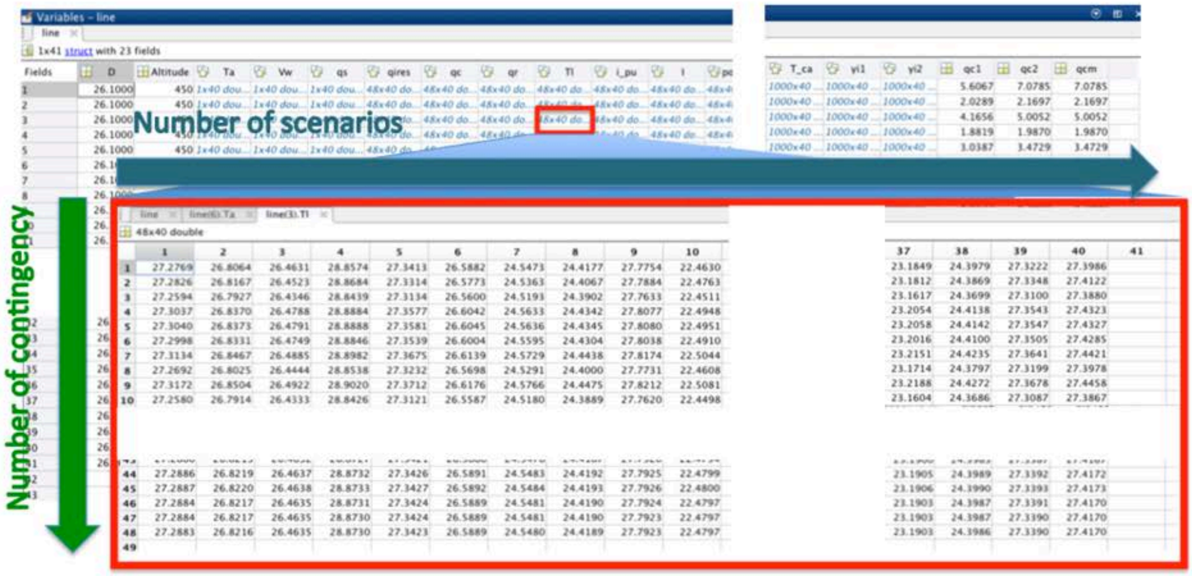


Figure A2. 3. Line structure zoom 3

Appendix 3: DeterministicNContigencies.m

```
clear all
clc
define_constants; %allows you to access the structure using a known name such as PD (real power demand), instead of having to remember that it's column 3
mpc=loadcase('case30');
mpopt = mpoption('PF_ALG', 2,'VERBOSE', 0, 'OUT_ALL', 0); %to avoid printing the results in the command window

%Calculation of the number of elements
Ngen = size(mpc.gencost,1);
Nbranch= size(mpc.branch,1);
Nbus= size(mpc.bus,1);

%Tl=temperature of the line
%Rl=resistence of the line
%qs=solar heat
%qc=convection heat
%qr=radiation heat

%Ambient temperature of each line [ºC]
% 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
Ta=[15 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25
25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25];

%Maximum temperature of the line [ºC]
Tl_max=120;

% Line conductor diameter of each line [mm]
% 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
D=[26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1
26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1
26.1 26.1 26.1 26.1];

% Wind speed at each line [m/s]
% 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
Vw=[15 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2];
%qs SOLAR HEAT of each line [W/m]
% 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
qs=[4.1 14.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 4.1 4.1 24.1 24.1
24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1 24.1
24.1 24.1 24.1];
```

```
%Altitude of each line [m]
%      1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20  21  22
23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
Altitude=[450  450  450  450  450  450  450  450  450  450  450  450  450  450  450  450  450
450  450  450  450  450  450  450  450  450  450  450  450  450  450  450  450  450  450  450
450  450  450  450  450];

%Heat Balance equation in steady state: I^2*R(Tl)+qs-qc-qr=0

%R(Tl) RESISTIVE LOSSES
Rref = 86e-6; %AC resistence 20ºC, units ohm/m
alpha = 0.0036; % (1/C)
Tref=20; % (C)
%Rl=Rref*(1+alpha*(Tl-Tref));

%qr RADIATION HEAT
epsilon = 0.5;
%qr=Alr*(((Tl+273)/100)^4-((Ta+273)/100)^4);

%qc CONVECTION
phi = 22.5 * pi/180; % wind angle
kang = 1.194 - cos(phi)+0.194*cos(2*phi) + 0.368 * sin(2*phi);

for i=1:Nbranch
   Tc(:,i) = linspace(Ta(i),Tl_max,1000);
   Tf(:,i) = (Tc(:,i) + Ta(i)) / 2;
   muf(:,i) = (1.458 * 1e-6 * (Tf(:,i)+273).^1.5)./(Tf(:,i) + 383.4);
   rof(:,i) = ( 1.293 - (1.525 * 1e-4 * Altitude(i))  + (6.379 * 1e-9 * (Altitude(i))^2) ) ./ (1 +
0.00367 .* Tf(:,i));
   kf(:,i)= 2.424 * 1e-2 + 7.477 * 1e-5 * Tf(:,i) - 4.407 * 1e-9 * (Tf(:,i)).^2;

   T_ca(:,i) = Tc(:,i) -Ta(i);

   %%%%%%%%%%%%%%%%%%%
   % at low wind speed

   yi1(:,i) = (1.01 + 0.0372 * ((D(i)*Vw(i)*rof(:,i)./muf(:,i)).^(0.52)))*kang .* kf(:,i) .* T_ca(:,i);
   [fitT1, gofT1, outputT1]= fit(T_ca(:,i),yi1(:,i),'poly1');
   qc1(i) = fitT1.p1;

   %%%%%%%%%%%%%%%%%%%
   % at high wind speed
   yi2(:,i) = 0.0119 * (D(i)*Vw(i)*rof(:,i)./muf(:,i)).^(0.6) .* kf(:,i) * kang .* T_ca(:,i);
   [fitT2, gofT2, outputT2]= fit(T_ca(:,i),yi2(:,i),'poly1');
   qc2(i) = fitT2.p1;

   qcm(i) = max (qc1(i),qc2(i));

   %qr RADIATION HEAT
```

```
   Alr(i)=0.0178 * D(i) * epsilon;
end

Reserve=0;
Conting=0;

%Save normal flow
results=runpf(mpc,mpopt); %save the results

for i = 1 : Nbranch
       branch_flow(Conting+1,i)=max(abs(results.branch(i,PF)),abs(results.branch(i,PT))); % we
always take the higher value of power injected "from" or to the "bus"
       Voltages(Conting+1,:)=results.bus(:,VM);
       if mpc.branch(i,BR_STATUS)==0
          I(Conting+1,i)=0;
          i_pu(Conting+1,i)=0;
       else

i_pu(Conting+1,i)=abs(((results.bus((mpc.branch(i,F_BUS)),VM)*exp(i*results.bus((mpc.branch
(i,F_BUS)),VA)*pi/180)-
results.bus((mpc.branch(i,T_BUS)),VM)*exp(i*results.bus((mpc.branch(i,T_BUS)),VA)*pi/180)))
/(mpc.branch(i,BR_R)+i*mpc.branch(i,BR_X)));

I(Conting+1,i)=(mpc.bus((mpc.branch(i,F_BUS)),BASE_KV))^2/mpc.branch(i,RATE_A).*i_pu(Con
ting+1, i);
       end
       options = optimset('TolX',1e-5);
       Tl(Conting+1,i)=(fzero(@(x)(Rref*(1+alpha*(x-Tref))*I(Conting+1,i).^2 + qs(i) ...
          - (Vw(i) == 0)*( 0.0205 * ( ( 1.293 - (1.525 * 1e-4 * Altitude(i))  + (6.379 * 1e-9 *
Altitude(i)^2) ) / (1 + 0.00367 * ((x + Ta(i))/2)) )^0.5 * D(i)^0.75 * (x-Ta(i))^1.25 ) ...
          - (Vw(i) ~= 0)*(qcm(i) * (x - Ta(i))) ...
          - (Alr(i)*(((x+273)/100)^4 - ((Ta(i)+273)/100)^4 ))),[Ta(i) 140],options));
end

%Simulation N-1 Analysis

%N-1 Generators Analysis
for N = 1 : Ngen
   Conting=N;
   mpc.gen(N,GEN_STATUS)=0; %default generator N, status=0

   for i=1 :Ngen
     if i~=N
        Reserve=Reserve+mpc.gen(i,PMAX)-mpc.gen(i,PG);
     end
   end

   DeltaPg(:,N)=((mpc.gen(:,PMAX)-mpc.gen(:,PG)).*mpc.gen(N,PG)/Reserve);
   DeltaPg(N,N)=0;
```

```
    mpc.gen(N,PG)=0;
    mpc.gen(:,PG)=mpc.gen(:,PG)+DeltaPg(:,N);

    results=runpf(mpc,mpopt); %save the results
    Voltages(Conting+1,:)=results.bus(:,VM);
    for i = 1 : Nbranch
        branch_flow(Conting+1,i)=max(abs(results.branch(i,PF)),abs(results.branch(i,PT))); % we
always take the higher value of power injected "from" or to the "bus"
        if mpc.branch(i,BR_STATUS)==0
            I(Conting+1,i)=0;
            i_pu(Conting+1, i)=0;
        else

i_pu(Conting+1,i)=abs(((results.bus((mpc.branch(i,F_BUS)),VM)*exp(i*results.bus((mpc.branch
(i,F_BUS)),VA)*pi/180)-
results.bus((mpc.branch(i,T_BUS)),VM)*exp(i*results.bus((mpc.branch(i,T_BUS)),VA)*pi/180)))
/(mpc.branch(i,BR_R)+i*mpc.branch(i,BR_X)));

I(Conting+1,i)=(mpc.bus((mpc.branch(i,F_BUS)),BASE_KV))^2/mpc.branch(i,RATE_A).*i_pu(Con
ting+1, i);
        end
        options = optimset('TolX',1e-5);
        Tl(Conting+1,i)=(fzero(@(x)(Rref*(1+alpha*(x-Tref))*I(Conting+1,i).^2 + qs(i) ...
            - (Vw(i) == 0)*( 0.0205 * ( ( 1.293 - (1.525 * 1e-4 * Altitude(i))  + (6.379 * 1e-9 *
Altitude(i)^2) ) / (1 + 0.00367 * ((x + Ta(i))/2)) )^0.5 * D(i)^0.75 * (x-Ta(i))^1.25 ) ...
            - (Vw(i) ~= 0)*(qcm(i) * (x - Ta(i))) ...
            - (Alr(i)*(((x+273)/100)^4 - ((Ta(i)+273)/100)^4 ))),[Ta(i) 140],options));
    end
    Reserve=0;
    mpc=loadcase('case30');
end

% N-1 Branch Analysis
for N = 1 : Nbranch
    Conting=Ngen+N
    mpc.branch(N,BR_STATUS)=0; %default line, status=0

    results=runpf(mpc,mpopt); %save the results
    Voltages(Conting+1,:)=results.bus(:,VM);

    for i = 1 : Nbranch

        branch_flow(Conting+1,i)=max(abs(results.branch(i,PF)),abs(results.branch(i,PT))); % we
always take the higher value of power injected "from" or to the "bus"

        if mpc.branch(i,BR_STATUS)==0
            I(Conting+1,i)=0;
            i_pu(Conting+1,i)=0;
        else
```

```
i_pu(Conting+1,i)=abs(((results.bus((mpc.branch(i,F_BUS)),VM)*exp(i*results.bus((mpc.branch
(i,F_BUS)),VA)*pi/180)-
results.bus((mpc.branch(i,T_BUS)),VM)*exp(i*results.bus((mpc.branch(i,T_BUS)),VA)*pi/180)))
/(mpc.branch(i,BR_R)+i*mpc.branch(i,BR_X)));

I(Conting+1,i)=(mpc.bus((mpc.branch(i,F_BUS)),BASE_KV))^2/mpc.branch(i,RATE_A).*i_pu(Con
ting+1, i);
        end
        options = optimset('TolX',1e-5);
        Tl(Conting+1,i)=(fzero(@(x)(Rref*(1+alpha*(x-Tref))*I(Conting+1,i).^2 + qs(i) ...
            - (Vw(i) == 0)*( 0.0205 * ( ( 1.293 - (1.525 * 1e-4 * Altitude(i))  + (6.379 * 1e-9 *
Altitude(i)^2) ) / (1 + 0.00367 * ((x + Ta(i))/2)) )^0.5 * D(i)^0.75 * (x-Ta(i))^1.25 ) ...
            - (Vw(i) ~= 0)*(qcm(i) * (x - Ta(i))) ...
            - (Alr(i)*(((x+273)/100)^4 - ((Ta(i)+273)/100)^4 ))),[Ta(i) 140],options));
    end
    mpc.branch(N,BR_STATUS)=1;
end
```

Appendix 4: ProbabilisticNContingencies.m

```
clear all
clc
define_constants; %allows you to access the structure using a known name such as PD (real
power demand), instead of having to remember that it's column 3
mpc=loadcase('case30');
mpopt = mpoption('PF_ALG', 2,'VERBOSE', 0, 'OUT_ALL', 0); %to avoid printing the results in
the command window

%Calculation of the number of elements
Ngen = size(mpc.gencost,1);
Nbranch= size(mpc.branch,1);
Nbus= size(mpc.bus,1);

%Tl=temperature of the line
%Rl=resistence of the line
%qires=current heat or resistive losses
%qs=solar heat
%qc=convection heat
%qr=radiation heat

%Fix de number of scenarios/sample size.
N_scenarios=500;

%Fix the line which graphs you want to represent
L_graph=1;

%Fix the conservative weather parameters
%Ambient temperature of each line [ºC]
%     1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20  21  22  23
24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
Ta_cons=[30  30  30  30  30  30  30  30  30   30  30  30  30  30  30  30  30  30  30  30
30  30  30  30  30  30  30  30  30  30  30  30  30  30  30  30  30  30  30  30  30];

% Wind speed at each line [m/s]
%      1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20  21  22
23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
Vw_cons=[0.5  0.5 0.5 0.5 0.5 0.5 0.5 0.5  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5];
%qs SOLAR HEAT of each line [W/m]
%     1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20  21  22  23
24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
qs_cons=[14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1
14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1 14.1
14.1 14.1 14.1 14.1 14.1];
```

```
%Fix the wind farm bus
Windfarm_gen_pos=5;
N_Windfarm=13;
Pr_Windturbine=2; %rated power on each Windturbine [MW]
%Windfarm_bus=2;
vi=3.5; % cut-in wind speed [m/s]
vr=12; % rated output wind speed [m/s]
v0=20; % cut-out wind speed [m/s]
a=Pr_Windturbine*vi^3/(vi^3-vr^3);
b=Pr_Windturbine/(vr^3-vi^3);

%Ambient temperature of each line [ºC]
%    1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23
24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
Ta_ave=[15  25  19  25  25  20  25  25  27  23  14  25  25  25  25  25  25  25  25  25  25
25  25  25  25  25  25  25  25  25  25  25  25  25  25  25  25  25  25  25  25];
Ta_std=[2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2];
%Maximum temperature of the line [ºC]
Tl_max=120;

% Line conductor diameter of each line [mm]
%  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23
24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
D=[26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1
26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1 26.1
26.1 26.1 26.1 26.1];

% Wind speed at each line [m/s]
%     1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23
24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
Vw_scale=[6  2  3  3  5  3  2  4  2.5  2  6  2  3  4  5  3  2  4  2.5  2  6  2  3  1
5  3  2  4  2.5  2  6  2  3  1  5  3  2  4  2.5  2  3.5];
Vw_shape=[2.5 1.5  3.5  1.8  2.3  1.7  2.4  2.2 1.5  2.5   2.5  2.5  2.4  3.3  2.7  2.5 1.5  3.5   1.8
2.3  1.7  2.4  2.2 1.5  2.5   2.5  2.5  2.4  3.3  2.7  2.5 1.5  3.5   1.8  2.3  1.7  2.4  2.2  1.5  2.5   2.5];
% Wind speed at the Windfarm
Vw_Windfarm_scale=8;
Vw_Windfarm_shape=3.6;
Vw_Windfarm=wblrnd(Vw_Windfarm_scale,Vw_Windfarm_shape,N_scenarios,1);

%qs SOLAR HEAT of each line [W/m]
%    1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23
24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
qs_ave=[11  11  8.1  9.1  9.5  10.1  8   9  10  9.1  12.1  9.8 10.3  11  11  8.1  9.1  9.5  10.1  8   9
10  9.1  12.1  9.8 10.3  11   11  8.1  9.1  9.5  10.1  8   9  10  9.1  12.1  9.8 10.3 9   11];
qs_std=[2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2];
%Altitude of each line [m]
```

```
%      1   2   3   4   5   6   7   8   9   10   11   12   13   14   15   16   17   18   19   20   21   22
23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
Altitude=[450 450 450 450 450 450 450 450 450 450 450 450 450 450 450 450 450
450 450 450 450 450 450 450 450 450 450 450 450 450 450 450 450 450 450 450
450 450 450 450 450];

%Heat Balance equation in steady state: I^2*R(Tl)+qs-qc-qr=0

%RESISTIVE LOSSES
Rref = 86e-6; %AC resistence 20ºC, units ohm/m
alpha = 0.0036; % (1/C)
Tref=20; % (C)
%Rl=Rref*(1+alpha*(Tl-Tref));

%qr RADIATION HEAT
epsilon = 0.5;
%qr=Alr*(((Tl+273)/100)^4-((Ta+273)/100)^4);

%qc CONVECTION
phi = 22.5 * pi/180; % wind angle
kang = 1.194 - cos(phi)+0.194*cos(2*phi) + 0.368 * sin(2*phi);

%Preallocation
Vw=zeros(N_scenarios,Nbranch);
Ta=zeros(N_scenarios,Nbranch);
qs=zeros(N_scenarios,Nbranch);
bus_pd=zeros(N_scenarios,Nbus);
bus_qd=zeros(N_scenarios,Nbus);
Voltages=zeros(Nbranch+Ngen+1,Nbus);
total_demand=zeros(N_scenarios,1);
rand_pdqd=zeros(N_scenarios,1);
total_demand=zeros(N_scenarios,1);
total_generation=zeros(N_scenarios,1);
total_capacity=zeros(N_scenarios,1);
generation_margin=zeros(N_scenarios,1);
deltaPG=zeros(N_scenarios,1);

for i=1:Nbranch
   Ta(:,i)=lhsnorm(Ta_ave(i),Ta_std(i),N_scenarios);
   Vw(:,i)=wblrnd(Vw_scale(i),Vw_shape(i),N_scenarios,1);
   qs(:,i)=abs(lhsnorm(qs_ave(i),qs_std(i),N_scenarios));

   line(i).D=D(i);
   line(i).Altitude=Altitude(i);
   line(i).Ta(1,:)=Ta(:,i);
   line(i).Vw(1,:)=Vw(:,i);
   line(i).qs(1,:)=qs(:,i);
```

```
    %qr RADIATION HEAT
    Alr(i)=0.0178*D(i)*epsilon;

    %Preallocation
    line(i).qires=zeros(Nbranch+Ngen+1,N_scenarios);
    line(i).qc=zeros(Nbranch+Ngen+1,N_scenarios);
    line(i).qr=zeros(Nbranch+Ngen+1,N_scenarios);
    line(i).Tl=zeros(Nbranch+Ngen+1,N_scenarios);
    line(i).i_pu=zeros(Nbranch+Ngen+1,N_scenarios);
    line(i).I=zeros(Nbranch+Ngen+1,N_scenarios);
    line(i).powerflow=zeros(Nbranch+Ngen+1,N_scenarios);
    line(i).Tc=zeros(1000,N_scenarios);
    line(i).Tf=zeros(1000,N_scenarios);
    line(i).muf=zeros(1000,N_scenarios);
    line(i).rof=zeros(1000,N_scenarios);
    line(i).kf=zeros(1000,N_scenarios);
    line(i).T_ca=zeros(1000,N_scenarios);
    line(i).yi1=zeros(1000,N_scenarios);
    line(i).yi2=zeros(1000,N_scenarios);
    line(i).qc1=zeros(1,N_scenarios);
    line(i).qc2=zeros(1,N_scenarios);
    line(i).qcm=zeros(1,N_scenarios);

    for j=1:N_scenarios
        line(i).Tc(:,j) = linspace(Ta(j,i),Tl_max,1000);
        line(i).Tf(:,j) = (line(i).Tc(:,j) + Ta(j,i)) / 2;
        line(i).muf(:,j) = (1.458 * 1e-6 * (line(i).Tf(:,j)+273).^1.5)./(line(i).Tf(:,j) + 383.4);
        line(i).rof(:,j) = ( 1.293 - (1.525 * 1e-4 * Altitude(i))  + (6.379 * 1e-9 * (Altitude(i))^2) ) ./ (1
+ 0.00367 .* line(i).Tf(:,j));
        line(i).kf(:,j)= 2.424 * 1e-2 + 7.477 * 1e-5 * line(i).Tf(:,j) - 4.407 * 1e-9 * (line(i).Tf(:,j)).^2;

        line(i).T_ca(:,j) = line(i).Tc(:,j) -Ta(j,i);

        %%%%%%%%%%%%%%%%%%%%
        % at low wind speed

        line(i).yi1(:,j) = (1.01 + 0.0372 * ((D(i)*Vw(j,i)*line(i).rof(:,j)./line(i).muf(:,j)).^(0.52)))*kang
.* line(i).kf(:,j) .* line(i).T_ca(:,j);
        [fitT1, gofT1, outputT1]= fit(line(i).T_ca(:,j),line(i).yi1(:,j),'poly1');
        line(i).qc1(1,j) = fitT1.p1;

        %%%%%%%%%%%%%%%%%%%%
        % at high wind speed
        line(i).yi2(:,j) = 0.0119 * (D(i)*Vw(j,i)*line(i).rof(:,j)./line(i).muf(:,j)).^(0.6) .* line(i).kf(:,j) *
kang .* line(i).T_ca(:,j);
        [fitT2, gofT2, outputT2]= fit(line(i).T_ca(:,j),line(i).yi2(:,j),'poly1');
        line(i).qc2(1,j) = fitT2.p1;

        line(i).qcm(1,j) = max (line(i).qc1(1,j),line(i).qc2(1,j));
```

```
    end
end

for i=1:Nbus
   rand_pdqd(:,1)=randn(N_scenarios,1);
   bus_pd(:,i)=mpc.bus(i,PD)+0.01*mpc.bus(i,PD).*rand_pdqd(:,1);
   bus_qd(:,i)=mpc.bus(i,QD)+0.01*mpc.bus(i,QD).*rand_pdqd(:,1);
end

%total_gen_capacity=0;
%for i=1:Ngen
    %if i~=Windfarm_gen_pos
        %total_gen_capacity=total_gen_capacity+mpc.gen(i,PMAX);
    %end
%end


for j=1:N_scenarios
    j
    Conting=0;

    if Vw_Windfarm(j)<vi
        mpc.gen(Windfarm_gen_pos,PG)=0;
    elseif ((Vw_Windfarm(j))>vi)&&((Vw_Windfarm(j))<vr)
        mpc.gen(Windfarm_gen_pos,PG)=N_Windfarm*(a+b*(Vw_Windfarm(j))^3);
    elseif (Vw_Windfarm(j)>vr)&&((Vw_Windfarm(j))<v0)
        mpc.gen(Windfarm_gen_pos,PG)=N_Windfarm*Pr_Windturbine;
    else
        mpc.gen(Windfarm_gen_pos,PG)=0;
    end

    Windfarm_PG(j)=mpc.gen(Windfarm_gen_pos,PG);

    for i=1:Nbus
        total_demand(j)=total_demand(j)+bus_pd(j,i);
    end

    for i=1:Ngen
       total_generation(j)=total_generation(j)+mpc.gen(i,PG);
       if i~=Windfarm_gen_pos
           total_capacity(j)=total_capacity(j)+mpc.gen(i,PMAX)-mpc.gen(i,PG);
           generation_margin(j)=generation_margin(j)+mpc.gen(i,PG)-mpc.gen(i,PMIN);
       end
    end

    deltaPG(j)=total_demand(j)-total_generation(j);

    for i=1:Ngen
        if deltaPG(j)>0
```

```
            if i~=Windfarm_gen_pos
                mpc.gen(i,PG)=mpc.gen(i,PG)+deltaPG(j)*((mpc.gen(i,PMAX)-
mpc.gen(i,PG))/total_capacity(j));
            end
        else
            if i~=Windfarm_gen_pos
                mpc.gen(i,PG)=mpc.gen(i,PG)+deltaPG(j)*((mpc.gen(i,PG)-
mpc.gen(i,PMIN))/generation_margin(j));
            end
        end
    end

    PG_gen(j,:)=mpc.gen(:,PG);

    for i=1:Nbus
        mpc.bus(i,PD)=bus_pd(j,i); %New demand at each node taking into account the
uncertainty for the load.
        mpc.bus(i,QD)=bus_qd(j,i);
    end

    results=runpf(mpc,mpopt);%save the results

    for i = 1 : Nbranch
        line(i).powerflow(Conting+1,j)=max(abs(results.branch(i,PF)),abs(results.branch(i,PT))); %
we always take the higher value of power injected "from" or to the "bus"
        Voltages(Conting+1,:)=results.bus(:,VM);
        if mpc.branch(i,BR_STATUS)==0
            line(i).I(Conting+1,j)=0;
            line(i).i_pu(Conting+1,j)=0;
        else

line(i).i_pu(Conting+1,j)=abs(((results.bus((mpc.branch(i,F_BUS)),VM)*exp(i*results.bus((mpc.
branch(i,F_BUS)),VA)*pi/180)-
results.bus((mpc.branch(i,T_BUS)),VM)*exp(i*results.bus((mpc.branch(i,T_BUS)),VA)*pi/180)))
/(mpc.branch(i,BR_R)+i*mpc.branch(i,BR_X)));

%line(i).I(Conting+1,j)=(mpc.bus((mpc.branch(i,F_BUS)),BASE_KV))^2/mpc.branch(i,RATE_A).*li
ne(i).i_pu(Conting+1,j);

line(i).I(Conting+1,j)=(mpc.branch(i,RATE_A)*1e6/sqrt(3)/(mpc.bus((mpc.branch(i,F_BUS)),BAS
E_KV)*1e3)).*line(i).i_pu(Conting+1,j);
        end
        options = optimset('TolX',1e-5);
        line(i).Tl(Conting+1,j)= (fzero(@(x)(Rref*(1+alpha*(x-Tref))*line(i).I(Conting+1,j).^2 + qs(j,i)
...
            - (Vw(j,i) == 0)*( 0.0205 * ( ( 1.293 - (1.525 * 1e-4 * Altitude(i))  + (6.379 * 1e-9 *
Altitude(i)^2) ) / (1 + 0.00367 * ((x + Ta(j,i))/2)) )^0.5 * D(i)^0.75 * (x-Ta(j,i))^1.25 ) ...
            - (Vw(j,i) ~= 0)*( line(i).qcm(1,j) * (x - Ta(j,i))) ...
            - (Alr(i)*(((x+273)/100)^4 - ((Ta(j,i)+273)/100)^4 ))),[Ta(j,i) 160],options));
```

```
        line(i).qires(Conting+1,j)=Rref*(1+alpha*(line(i).Tl(Conting+1,j)-
Tref))*line(i).I(Conting+1,j).^2;
        line(i).qc(Conting+1,j)=(Vw(j,i) == 0)*( 0.0205 * ( ( 1.293 - (1.525 * 1e-4 * Altitude(i))  +
(6.379 * 1e-9 * Altitude(i)^2) ) / (1 + 0.00367 * ((line(i).Tl(Conting+1,j) + Ta(j,i))/2)) )^0.5 *
D(i)^0.75 * (line(i).Tl(Conting+1,j)-Ta(j,i))^1.25 ) ...
            + (Vw(j,i) ~= 0)*( line(i).qcm(1,j) * (line(i).Tl(Conting+1,j) - Ta(j,i)));
        line(i).qr(Conting+1,j)=(Alr(i)*(((line(i).Tl(Conting+1,j)+273)/100)^4 - ((Ta(j,i)+273)/100)^4
));
    end

    Reserve=0;
    %Simulation N-1 Analysis

    %N-1 Generators Analysis
    for N = 1 : Ngen
        Conting=N;
        mpc.gen(N,GEN_STATUS)=0; %default generator N, status=0

        for i=1 :Ngen
            if i~=N
                Reserve=Reserve+mpc.gen(i,PMAX)-mpc.gen(i,PG);
            end
        end

        DeltaPg(:,N)=((mpc.gen(:,PMAX)-mpc.gen(:,PG)).*mpc.gen(N,PG)/Reserve);
        DeltaPg(N,N)=0;
        mpc.gen(N,PG)=0;
        mpc.gen(:,PG)=mpc.gen(:,PG)+DeltaPg(:,N);

        results=runpf(mpc,mpopt); %save the results
        Voltages(Conting+1,:)=results.bus(:,VM);

        for i = 1 : Nbranch
            line(i).powerflow(Conting+1,j)=max(abs(results.branch(i,PF)),abs(results.branch(i,PT)));
% we always take the higher value of power injected "from" or to the "bus"
            Voltages(Conting+1,:)=results.bus(:,VM);
            if mpc.branch(i,BR_STATUS)==0
                line(i).I(Conting+1,j)=0;
                line(i).i_pu(Conting+1,j)=0;
            else

line(i).i_pu(Conting+1,j)=abs(((results.bus((mpc.branch(i,F_BUS)),VM)*exp(i*results.bus((mpc.
branch(i,F_BUS)),VA)*pi/180)-
results.bus((mpc.branch(i,T_BUS)),VM)*exp(i*results.bus((mpc.branch(i,T_BUS)),VA)*pi/180)))
/(mpc.branch(i,BR_R)+i*mpc.branch(i,BR_X)));

%line(i).I(Conting+1,j)=(mpc.bus((mpc.branch(i,F_BUS)),BASE_KV))^2/mpc.branch(i,RATE_A).*li
ne(i).i_pu(Conting+1,j);
```

```
line(i).I(Conting+1,j)=(mpc.branch(i,RATE_A)*1e6/sqrt(3)/(mpc.bus((mpc.branch(i,F_BUS)),BAS
E_KV)*1e3)).*line(i).i_pu(Conting+1,j);
            end
            options = optimset('TolX',1e-5);
            line(i).Tl(Conting+1,j)= (fzero(@(x)(Rref*(1+alpha*(x-Tref))*line(i).I(Conting+1,j).^2 +
qs(j,i) ...
                - (Vw(j,i) == 0)*( 0.0205 * ( ( 1.293 - (1.525 * 1e-4 * Altitude(i))  + (6.379 * 1e-9 *
Altitude(i)^2) ) / (1 + 0.00367 * ((x + Ta(j,i))/2)) )^0.5 * D(i)^0.75 * (x-Ta(j,i))^1.25 ) ...
                - (Vw(j,i) ~= 0)*( line(i).qcm(1,j) * (x - Ta(j,i))) ...
                - (Alr(i)*(((x+273)/100)^4 - ((Ta(j,i)+273)/100)^4 ))),[Ta(j,i) 160],options));
            line(i).qires(Conting+1,j)=Rref*(1+alpha*(line(i).Tl(Conting+1,j)-
Tref))*line(i).I(Conting+1,j).^2;
            line(i).qc(Conting+1,j)=(Vw(j,i) == 0)*( 0.0205 * ( ( 1.293 - (1.525 * 1e-4 * Altitude(i))  +
(6.379 * 1e-9 * Altitude(i)^2) ) / (1 + 0.00367 * ((line(i).Tl(Conting+1,j) + Ta(j,i))/2)) )^0.5 *
D(i)^0.75 * (line(i).Tl(Conting+1,j)-Ta(j,i))^1.25 ) ...
                + (Vw(j,i) ~= 0)*( line(i).qcm(1,j) * (line(i).Tl(Conting+1,j) - Ta(j,i)));
            line(i).qr(Conting+1,j)=(Alr(i)*(((line(i).Tl(Conting+1,j)+273)/100)^4 -
((Ta(j,i)+273)/100)^4 ));
        end
        Reserve=0;
        mpc.gen(N,GEN_STATUS)=1;
        mpc.gen(:,PG)=PG_gen(j,:);
    end

    % N-1 Branch Analysis
    for N = 1 : Nbranch
        Conting=Ngen+N;
        mpc.branch(N,BR_STATUS)=0; %default line, status=0

        results=runpf(mpc,mpopt); %save the results
        Voltages(Conting+1,:)=results.bus(:,VM);

        for i = 1 : Nbranch
            line(i).powerflow(Conting+1,j)=max(abs(results.branch(i,PF)),abs(results.branch(i,PT)));
% we always take the higher value of power injected "from" or to the "bus"
            Voltages(Conting+1,:)=results.bus(:,VM);
            if mpc.branch(i,BR_STATUS)==0
                line(i).I(Conting+1,j)=0;
                line(i).i_pu(Conting+1,j)=0;
            else

line(i).i_pu(Conting+1,j)=abs(((results.bus((mpc.branch(i,F_BUS)),VM)*exp(i*results.bus((mpc.
branch(i,F_BUS)),VA)*pi/180)-
results.bus((mpc.branch(i,T_BUS)),VM)*exp(i*results.bus((mpc.branch(i,T_BUS)),VA)*pi/180)))
/(mpc.branch(i,BR_R)+i*mpc.branch(i,BR_X)));

%line(i).I(Conting+1,j)=(mpc.bus((mpc.branch(i,F_BUS)),BASE_KV))^2/mpc.branch(i,RATE_A).*li
ne(i).i_pu(Conting+1,j);
```

```
line(i).I(Conting+1,j)=(mpc.branch(i,RATE_A)*1e6/sqrt(3)/(mpc.bus((mpc.branch(i,F_BUS)),BAS
E_KV)*1e3)).*line(i).i_pu(Conting+1,j);
        end
        options = optimset('TolX',1e-5);
        line(i).Tl(Conting+1,j)= (fzero(@(x)(Rref*(1+alpha*(x-Tref))*line(i).I(Conting+1,j).^2 +
qs(j,i) ...
          - (Vw(j,i) == 0)*( 0.0205 * ( ( 1.293 - (1.525 * 1e-4 * Altitude(i))  + (6.379 * 1e-9 *
Altitude(i)^2) ) / (1 + 0.00367 * ((x + Ta(j,i))/2)) )^0.5 * D(i)^0.75 * (x-Ta(j,i))^1.25 ) ...
          - (Vw(j,i) ~= 0)*( line(i).qcm(1,j) * (x - Ta(j,i))) ...
          - (Alr(i)*(((x+273)/100)^4 - ((Ta(j,i)+273)/100)^4 ))),[Ta(j,i) 170],options));
        line(i).qires(Conting+1,j)=Rref*(1+alpha*(line(i).Tl(Conting+1,j)-
Tref))*line(i).I(Conting+1,j).^2;
        line(i).qc(Conting+1,j)=(Vw(j,i) == 0)*( 0.0205 * ( ( 1.293 - (1.525 * 1e-4 * Altitude(i))  +
(6.379 * 1e-9 * Altitude(i)^2) ) / (1 + 0.00367 * ((line(i).Tl(Conting+1,j) + Ta(j,i))/2)) )^0.5 *
D(i)^0.75 * (line(i).Tl(Conting+1,j)-Ta(j,i))^1.25 ) ...
          + (Vw(j,i) ~= 0)*( line(i).qcm(1,j) * (line(i).Tl(Conting+1,j) - Ta(j,i)));
        line(i).qr(Conting+1,j)=(Alr(i)*(((line(i).Tl(Conting+1,j)+273)/100)^4 -
((Ta(j,i)+273)/100)^4 ));
      end
      mpc.branch(N,BR_STATUS)=1;
   end
end
```