



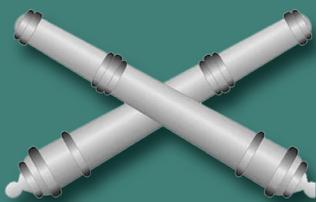
ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



MANTENIMIENTO BASADO EN CONFIABILIDAD

Introducción a las Matemáticas
con Aplicaciones en Python

Alexandre Galo Lopes



CRÉDITOS

MANTENIMIENTO BASADO EN CONFIABILIDAD

Introducción a las Matemáticas con Aplicaciones en Python

Alexandre Galo Lopes

ISBN: 978-9942-765-61-1

Junio de 2020

Pares revisión científica

Ing. Jose Antonio Toscano Romero (Universidad de las Américas – UDLA)

Ing. Omar Cristóbal Flor Unda (Universidad de las Américas – UDLA)

Universidad de las Fuerzas Armadas ESPE

Tern. Humberto Anibal Parra Cárdenas, Ph.D. (Rector)

Publicación autorizada por:

Comisión Editorial de Libros de la Universidad de las Fuerzas Armadas ESPE

Edición y producción:

Lcdo. Pedro Xavier Chinga Marmol

pxchinga@espe.edu.ec

Derechos reservados. Se prohíbe la reproducción de esta obra por cualquier medio impreso, reprográfico o electrónico.

El contenido, uso de fotografías, gráficos, cuadros, tablas y referencias es de exclusiva responsabilidad del autor.

Universidad de las Fuerzas Armadas ESPE

Av. General Rumiñahui s/n, Sangolquí, Ecuador

www.espe.edu.ec

Los derechos de esta edición electrónica son de la Universidad de las Fuerzas Armadas ESPE, para consulta de profesores y estudiantes de la universidad e investigadores en www.repositorio.espe.edu.ec.



Índice

1	Introducción	5
1.1	Motivación	5
1.2	Breve histórico de la <i>Reliability Centered Mantainance (RCM)</i>	6
1.3	Fallas	7
1.3.1	Características de la falla	7
1.4	Confiabilidad	7
1.5	Leyes de fallas	8
1.6	Ciclo de Vida	9
1.6.1	Costos en el ciclo de vida	11
2	Conceptos Básicos de Probabilidades	12
2.1	Probabilidades de ocurrencia	12
2.2	Distribuciones de probabilidades	12
2.2.1	Parámetros de localización y de escala	20
2.2.2	Principales estadísticas	22
2.2.3	Distribución normal	23
2.2.4	Distribución uniforme	25
2.2.5	Distribución de Weibull	27
2.2.6	Distribución beta	29
2.2.7	Distribución binomial	31
2.2.8	Distribución de Poisson	35
2.3	Estimación de parámetros y ajuste de distribuciones	37
2.3.1	Criterios para estimativas	38
2.3.2	Método de la máxima similitud	42
2.3.3	Método de los mínimos cuadrados	43
2.3.4	Test de Kolmogorov-Smirnov	45
2.3.5	Criterio de información de Akaike	47

3	Ingeniería de Confiabilidad	50
3.1	Cálculos de confiabilidad	50
3.1.1	Tasa de reparos o ROCOF	53
3.1.2	Técnicas de análisis exploratorio de datos	56
3.1.3	Confiabilidad de sistemas	68
4	Tipos de Mantenimiento	74
4.1	Mantenimiento correctivo	74
4.1.1	Medidas de mantenimiento correctivo	75
4.1.2	Modelos matemáticos de mantenimiento correctivo	76
4.2	Mantenimiento preventivo	85
4.2.1	Modelos matemáticos de mantenimiento preventivo	87
4.3	Mantenimiento predictivo	92
4.4	Mantenimiento modificador	94
4.4.1	Confiabilidad creciente	95
4.4.2	Modelos de aceleración física	99
5	Operaciones de Mantenimiento	101
5.1	Mantenibilidad	101
5.1.1	Medidas de mantenibilidad	101
5.1.2	Factores humanos	102
5.2	Control de stock	107
5.2.1	Modelo de cantidad económica de compra	108
5.2.2	Modelo de cantidad de pedidos para la producción	110
5.2.3	Modelo de pedidos con descuento	111
5.2.4	Stock de seguridad	112
5.2.5	Disponibilidad de piezas en stock	113
5.3	Costos de mantenimiento	113
5.3.1	Relación costo beneficio	114
5.4	Control del mantenimiento	116
6	Aprendizaje de Máquina	118
6.1	Nociones básicas	119
6.1.1	Clasificación de parámetros de sistema	119
6.1.2	Redes neuronales artificiales	129



1. Introducción

1.1 Motivación

La Ingeniería de Mantenimiento es la actividad de mantenimiento de un ítem/equipamiento que desarrolla conceptos, criterios y requisitos técnicos en las fases conceptual y de adquisición para ser usado en un corriente *status* durante la fase operativa para garantizar un efectivo soporte de mantenimiento del equipo. [DOD (1968)] *apud* [Dhillon (2002)]

El Mantenimiento Basado en Confiabilidad (MBC) (adaptado del inglés *Reliability Centered Maintenance* (RCM)) es el proceso usado para determinar un abordaje más efectivo para el mantenimiento. Involucra la identificación de acciones que, cuando son adoptadas, reducirán la probabilidad de falla y tendrán un costo más adecuado. [NASA (2000)]

Este libro tiene como reto presentar algunos de los principales modelos matemáticos y estadísticos de interés para el MBC, cuyas aplicaciones básicas en lenguaje computacional Python [PSF (2018)] permitirán al lector desarrollar sus propias aplicaciones de control de mantenimiento.

Las aplicaciones serán desarrolladas con las bibliotecas disponibles para Python, ya ampliamente difundidas y disponibles gratuitamente a los desarrolladores. El uso de las bibliotecas consolidadas minimiza la necesidad de validaciones de sus rutinas, una vez que el proceso de maduración de sus códigos ya es hecho por la comunidad usuaria del lenguaje alrededor del mundo.

Sin embargo, una vez comprendidas las técnicas matemáticas y las rutinas computacionales por el lector, le será posible migrar para otros lenguajes de su interés, perfeccionar los códigos presentados –todos libres y abiertos– adaptar las rutinas para importación de datos en tiempo real, tratar los resultados con otras herramientas, como las aplicaciones de *machine learning*, por ejemplo.

Este libro no tendrá la pretensión de agotar el asunto. Es necesario que el lector tenga en su mente que las bibliotecas y paquetes evolucionan a lo largo del tiempo, y que las rutinas presentadas deberán ser constantemente revisadas. Algunos de los parámetros pueden convertirse en obsoletos en versiones posteriores de las bibliotecas computacionales, o hasta mismo perder sus funcionalidades. Cabe destacar que la bibliografía adoptada para la base de conocimiento que dio origen a este libro, presenta una gran cantidad de datos que servirán de forma complementaria para la implementación de un eficiente sistema de mantenimiento, no siendo sustituible por ese

texto. Eso es una motivación para que el lector haga del conocimiento presentado una fuente básica, pero no estática. Hay que comprender los conceptos de manera dinámica, más por su filosofía y fundamentación matemática que por el texto publicado.

En cuanto a la adopción de la lenguaje Python para los códigos desarrollados en este libro, una motivación se presentó en el año de 2018 cuando el Instituto de Ingenieros, Electricistas y Electrónicos (IEEE), publicó el resultado de una investigación sobre popularidad de 47 lenguajes computacionales, en que el Python apareció en primer lugar [IEEE (2018)].

1.2 Breve histórico de la *Reliability Centered Maintenance* (RCM)

Según el guía de la NASA [NASA (2000)], el inicio del RCM ocurrió en los años de 1960 y 1970, cuando la industria de aviación desarrolló análisis de políticas de mantenimiento. El trabajo de [Nowlan y Heap (1978)] demostró que no existía una fuerte correlación entre la edad y la tasa de fallas de los equipos, lo que invalidaba la premisa básica de que el mantenimiento debería ser basado en el tiempo. [Nowlan y Heap (1978)] *apud* [NASA (2000)].

El mantenimiento preventivo, fue la técnica más avanzada empleada entre las décadas de 1960 a 1980 [NASA (2000)] y se basa en dos principios:

1. existe una fuerte correlación entre la edad del equipamiento y su tasa de fallas; y
2. la probabilidad de falla de un componente individual y del equipamiento puede ser determinada estadísticamente debido a la incertidumbre de las ocurrencias, lo que permite que las piezas puedan ser repuestas o reconstituidas antes de la falla.

Un proceso denominado Exploración de la Edad (del inglés *Age Exploration (AE)*) fue empleado por la Fuerza de Submarinos Estadounidense, en la década de 1970, para extender el tiempo entre mantenimientos completos (*overhauls*) y para sustituir las tareas de mantenimiento basado en el tiempo por mantenimientos basados en las condiciones del equipo. La Armada Estadounidense entonces invocó requisitos de RCM y monitoreo de condiciones como parte del diseño de especificaciones de nuevas embarcaciones [NASA (2000)].

Con la finalidad de obtenerse una estrategia óptima de mantenimiento, para optimizar la operabilidad y la eficiencia del equipo, minimizando el costo de su ciclo de vida, el programa de RCM puede ser basado en los siguientes componentes:

1. Mantenimiento Correctivo
 - Pequeños componentes
 - Piezas que no sean críticas
 - Piezas cuyos daños no traen serias consecuencias
 - Piezas cuya falla es improbable
 - Piezas redundantes
2. Mantenimiento Preventivo
 - Piezas sujetas a desgaste
 - Repuestos consumibles
 - Piezas con patrón de fallas conocido
3. Mantenimiento Predictivo
 - Piezas con patrón de falla al azar
 - Piezas no sometidas al desgaste
 - Piezas cuyas fallas fueron inducidas por mantenimiento preventivo
4. Mantenimiento Modificador
 - Resultado de análisis de causa raíz de las fallas
 - Resultado de exploración de la edad
 - Resultado de análisis de modo y efecto de la falla

1.3 Fallas

Para fines de mejor comprensión de este libro, se considera falla a una interrupción de la función del equipo, del componente o de la pieza, o la degradación substancial de su desempeño.

Es posible hacer un estudio sistemático de las fallas investigando el sistema, el subsistema, el componente o una pieza. Poseyendo el conocimiento respecto el origen de la falla, es posible concluir en qué nivel de comprometimiento el sistema se encuentra. La falla de una pieza no necesariamente implica en una falla del sistema; puede darse el caso de que esa pieza no sea fundamental al funcionamiento del sistema o posea una análoga en redundancia, por ejemplo. El ingeniero Stanley Parker trabajaba en una fábrica de torpedos de la Armada Británica y descubrió que piezas consideradas inadecuadas para la montaje de los torpedos, en verdad, funcionaban cuando eran montadas en conjunto, porque había una mala interpretación del sistema de tolerancias geométricas, lo que dio origen al sistema *Geometric Dimensioning & Tolerancing* (GD&T) [Wandeck (2010)] *apud* [Stoco *et al.* (2016)].

Al revés de eso, es posible que ocurra una falla del sistema, debido a un conjunto de componentes adecuados que no funcionan bien cuando son colocados juntos.

1.3.1 Características de la falla

Según [NASA (2000)], la curva de probabilidad condicional de falla (P_{cond} versus tiempo) fue sometida a tres estudios independientes, los cuales llegaron a la conclusión que los materiales se quedan según las proporciones presentadas en la Fig. 1.1.

Los estudios mencionados revelaron que las fallas al azar ocurrían en 77-92% del número de fallas totales y las fallas relacionadas a la edad de los equipos ocurrían en 8-23% [NASA (2000)].

Las fallas fueron clasificadas en seis tipos de patrones, a saber [NASA (2000)]:

- **Tipo A** - Probabilidad de falla constante o gradualmente creciente, seguida de una acentuada región de desgaste. Una edad límite puede ser deseable (falla típica de motores alternativos);
- **Tipo B** - Mortalidad infantil, seguida por una probabilidad de falla constante o de crecimiento despacio (falla típica de equipamientos electrónicos);
- **Tipo C** - Baja probabilidad de falla cuando el ítem es nuevo o recientemente recuperado a su estado de nuevo (*overhauled*), seguido por un rápido crecimiento para un nivel relativamente constante;
- **Tipo D** - Probabilidad de falla relativamente constante para todas las edades;
- **Tipo E** - Curva de la bañera; mortalidad infantil seguida de una probabilidad de falla constante o gradualmente creciente y entonces seguida de una región de desgaste pronunciado. Una edad límite puede ser deseable; y
- **Tipo F** - Probabilidad de falla gradualmente creciente, pero no es identificada una edad de desgaste. Una edad límite no es aplicable (falla típica de motores a turbinas).

Los tipos A y E son típicos de piezas individuales y componentes simples como neumáticos, álabes de compresores, pedales de freno y miembros estructurales [NASA (2000)]. Piezas simples frecuentemente demuestran una relación directa entre confiabilidad y edad.

1.4 Confiabilidad

Una vez que el objetivo del presente texto es el estudio del mantenimiento basado en la confiabilidad, es necesario definir, de antemano, su concepto.

La confiabilidad de un ítem (sistemas, subsistemas, componentes o piezas) es la probabilidad de que ese ítem funcione adecuadamente por un período de operación, bajo determinadas condiciones, sin presentar fallas.

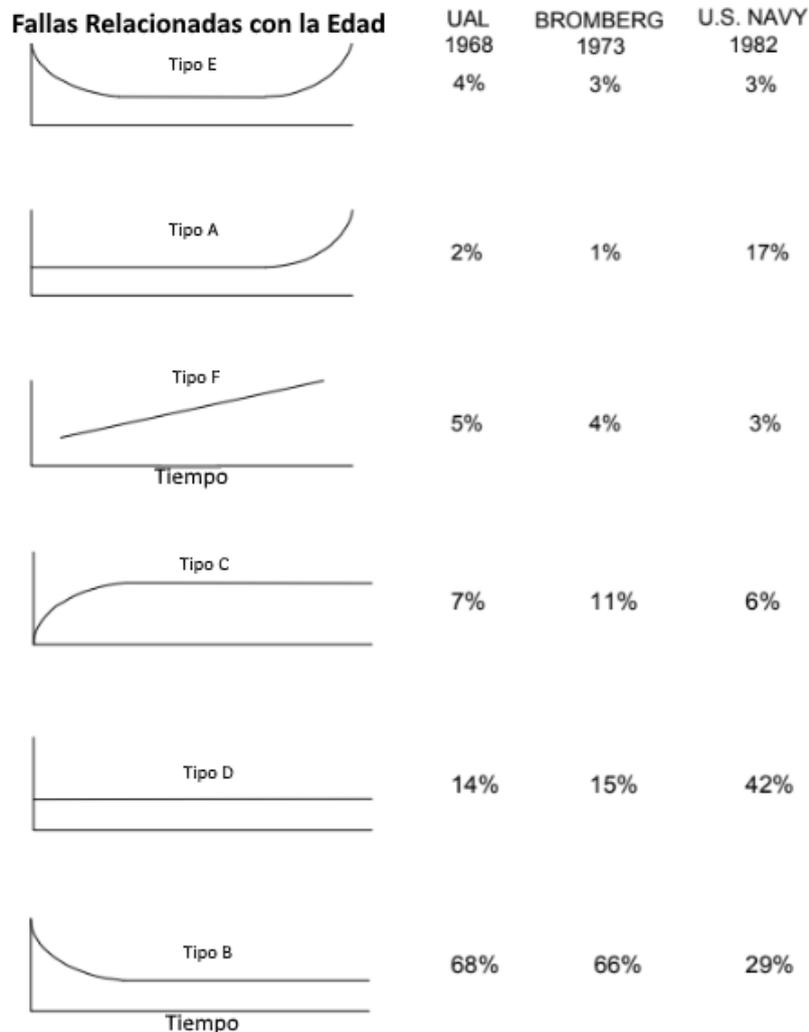


Figura 1.1. Curvas de Probabilidad Condicional de Fallas. Copyright 2000 por National Aeronautics and Space Administration.

La tasa de fallas es útil para calcular costos y la frecuencia de los mantenimientos, pero una solución de mantenimiento también debe considerar aspectos como seguridad, consecuencias económicas y eficiencia del programa de mantenimiento (reducción de las consecuencias de las fallas).

1.5 Leyes de fallas

Como ya se vio en la sección 1.4, la confiabilidad (y posteriormente serán presentados otros conceptos correlatos) es una probabilidad que puede ser bien modelada por una función matemática. Esa función generalmente se presenta como un modelo estadístico conocido, lo que permite al ingeniero de mantenimiento hacer buenas previsiones y estimativas que involucran tiempo hasta el próximo daño, costos de mantenimiento, necesidad de repuestos, fin de la vida útil del equipo.

Para comprender las leyes de distribución de fallas, serán presentados dos conceptos empleados en la literatura técnica de mantenimiento [NIST/SEMATECH (2012)]:

- **Sistema Reparable** - es aquel que puede ser restaurado a una operación satisfactoria por cualquier acción, incluyendo reemplazos de piezas o cambios a configuraciones ajustables.
- **Población no Reparable** - es aquella en que el ítem que falla es removido permanentemente de la población.

Para modelar la distribución de fallas, será presentada la Tabla 1.1 [DMAvEx (2010)].

1.6 Ciclo de Vida

Todo material presenta una vida útil, por lo que es sometido a una fase de desarrollo, a ciclos de mantenimiento, de modernización y a una fase de desecho. A eso se llama ciclo de vida del material.

El ciclo de vida de un material es definido como el intervalo de tiempo que tiene inicio con la entrada del material en servicio y finaliza con una intervención del tipo revitalización o modernización (o que es un marco para el inicio de un nuevo ciclo de vida) o con su desactivación. La entrada en servicio se refiere al primer ciclo de vida o a cualquier otro ciclo delimitado por intervenciones del tipo revitalización o modernización [DMAvEx (2010)]. La Fig. 1.2 ilustrará los diversos ciclos de vida de un material.

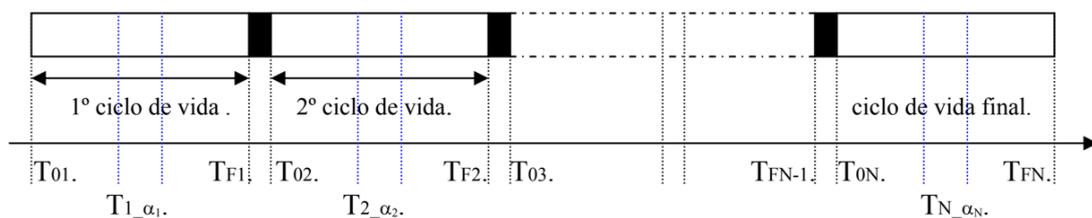


Figura 1.2. Esquema ilustrativo de los ciclos de vida de un material. Copyright 2010 por Diretoria de Material de Aviação do Exército

En la Fig. 1.2 se quedan definidos los siguientes términos:

1. T_{01} - fecha de entrada en servicio para el primer ciclo de vida (material recién adquirido);
2. T_{F1} - fecha de la primera intervención límite del ciclo de vida;
3. T_{02} - fecha de entrada en servicio para el segundo ciclo de vida;
4. $T_{F(M-1)} - T_{0M}$ - periodo de tiempo en que el material se encuentra sujeto a una intervención límite de ciclo de vida, donde la variable M indexa el tiempo ($2 \leq M \leq N$);
5. T_{0N} - fecha de entrada en servicio para el ciclo de vida final;
6. T_{FN} - fecha de desactivación del material;
7. $T_{Y\alpha_Y}$ - secuencia de intervenciones de ciclo continuado referentes al Y -ésimo ciclo de vida del material, donde $1 \leq Y \leq N$ y α_Y es una secuencia de números enteros que indexan las intervenciones de ciclo continuado (grandes inspecciones, por ejemplo) a lo largo del Y -ésimo ciclo de vida;
8. N - número de ciclos de vida a lo largo de la vida en servicio del material;
9. $T_{FN} - T_{01}$ - amplitud total de vida en servicio de un material, cuya información es suministrada por el fabricante, por la literatura técnica, por una base normativa, etc. Por ejemplo, considere una aeronave militar cuya célula, incluida la fuselaje, estructura y mecanismos, carenados, superficies de sustentación y control, permite una longevidad $T_{FN} - T_{01} = 35$ años con una intervención de modernización que resulta $N = 2$, o sea, con dos ciclos medios de 17,5 años.

Tabla 1.1

Ley de distribución de fallas por tipo de sistema, subsistema o componente

LEY DE DISTRIBUCIÓN DE FALLAS	SISTEMA/SUBSISTEMA/COMPONENTE
Normal	Sistemas o componentes mecánicos en los cuales la falla sea debida a algún efecto de desgaste (sistema con memoria y tasa de falla creciente)
LogNormal	Análisis de sistemas o componentes sujetos a fallas por fatiga, resistencia de los materiales, proyectos con cargas variables Sistemas con componentes reparables, los cuales demandan una atención especial en el cálculo del tiempo entre reparos (parcela de vida)
Exponencial	Sistemas o componentes electrónicos con tasa de falla constante, principalmente en aplicaciones militares. No contempla el efecto del desgaste (<i>memoryless</i>)
Binomial	Cálculo de la probabilidad del nivel de supervivencia de un sistema, por ocasión del análisis de un Diagrama de Bloques de Confiabilidad
Poisson	Sistemas o componentes sujetos a perturbaciones que pueden o no resultar en falla. Es un límite de la distribución binomial
Exponencial - Poisson	Sistemas o componentes sujetos a perturbaciones (Poisson-discretos), computadas durante un periodo de tiempo (Exponencial-continua), en que la falla ocurre si, y sólo si, al menos una perturbación ocurre Sistemas o componentes sujetos a perturbaciones (Poisson-discretos), con probabilidad constante de que ellos no resultan en fallas, computadas durante un periodo de tiempo (Exponencial-continua)
Gama	Sistemas o componentes sujetos a perturbaciones, en que la falla ocurre después de la ocurrencia de un número mínimo de perturbaciones. La duración hasta la falla, T, sigue la Ley de Fallas Gama
Gama - Poisson	Sistemas o componentes sujetos a perturbaciones (Poisson-discretos), computadas durante un periodo de tiempo (Exponencial-continua), en que la falla ocurre después de la ocurrencia de un número mínimo de perturbaciones
Erlang	Sistemas que consisten en la combinación de subsistemas exponenciales con la misma tasa de falla (es una simplificación algebraica)
Weibull	Sistema compuesto de varios componentes, con falla ocurriendo esencialmente debido a más grave imperfección o irregularidad, entre un grande número de imperfecciones del sistema

1.6.1 Costos en el ciclo de vida

Una de las finalidades básicas del mantenimiento es poner el material en estado de uso seguro y eficiente, de manera que bajen los costos de paradas inopinadas. El mantenimiento busca, más allá de mantener el material en funcionamiento, mantener sus costos de operación a niveles óptimamente bajos.

Los costos de operación y de mantenimiento de un sistema genérico pueden estar bien representados por la Fig. 1.3

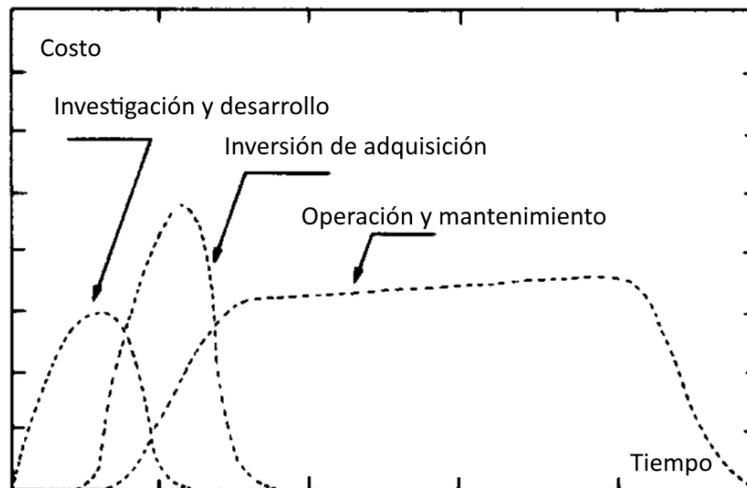


Figura 1.3. Relación general de costo y fase para un sistema. Copyright 1981 por Pergamon Press Ltd

Para la NASA [NASA (2000)], la fase de planificación (incluso el diseño conceptual) consume 2/3 de todo el costo del ciclo de vida de una instalación. Las fases subsecuentes de planificación consumen 29% del costo del ciclo de vida y un 5% del costo del ciclo de vida puede verse afectado por las fases finales del ciclo. La Fig. 1.4 presenta esas distribuciones acumuladas.

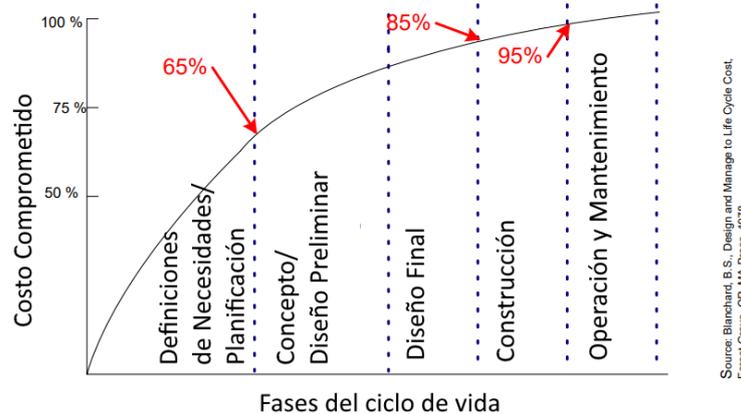


Figura 1.4. Relación general de costo planificado y fase para un sistema. Copyright 2004 por National Aeronautics and Space Administration



2. Conceptos Básicos de Probabilidades

2.1 Probabilidades de ocurrencia

Un sistema sometido a condiciones diversas de funcionamiento puede también presentar diversas posibilidades de fallas, en una parte significativa de las veces, imprevisibles bajo el punto de vista determinista. Pero hay fallas que ocurren con mayor o con menor frecuencia, cuyos comportamientos pueden ser modelados matemáticamente con el uso de herramientas estadísticas, suministrando una previsibilidad bajo el punto de vista probabilista.

Con la finalidad de establecerse un criterio de frecuencia de fallas, con vistas a un análisis de efectos de las mismas sobre el comportamiento del sistema, un experto puede cuantificar la probabilidad de fallas con informaciones del histórico de ocurrencias en el sistema, del histórico de ocurrencias en sistemas semejantes, o con base en una suposición inicial, ajustada periódicamente a partir del aprendizaje con el comportamiento de fallas del mismo sistema.

La Tabla 2.1 [NASA (2000)] presentará una sugerencia de clasificación de probabilidad de fallas, con base en la frecuencia de sus ocurrencias. Esa sugerencia puede ser adaptada para aplicaciones específicas donde, por ejemplo, el experto necesite una cantidad mayor o menor de clasificaciones. La frecuencia de las ocurrencias puede presentarse en horas de operación, días, ciclos o cualquier otro abordaje métrico considerado adecuado.

2.2 Distribuciones de probabilidades

Conforme lo presentado por [NIST/SEMATECH (2012)], una distribución de probabilidad es un modelaje matemático que permite:

- Calcular intervalos de confianza para parámetros y para calcular regiones críticas para test de hipótesis;
- Para datos univariados, es frecuentemente útil para determinar el modelo distribucional razonable para esos datos;
- Los intervalos estadísticos y el test de hipótesis son frecuentemente basados en premisas distribucionales específicas. Antes de computar un intervalo o test basado en una premisa distribucional, es necesario verificar que la premisa es justificada para un determinado conjunto de datos. En este caso, la distribución no necesita ser el mejor ajuste para los

datos, pero un adecuado y suficiente modelo tal que las técnicas estadísticas permitan conclusiones válidas;

- Estudios de simulaciones con números al azar generados para ser usados con una distribución de probabilidad específica son frecuentemente necesarios.

Tabla 2.1

Clasificación de fallas según sus ocurrencias

NIVEL	FRECUENCIA	IDENTIFICACIÓN
1	1/10000	Probabilidad remota de ocurrencia; expectativa de ocurrencia no razonable
2	1/5000	Baja tasa de fallas; similar al diseño anterior que, en el pasado, tenía bajas tasas de fallas para ciertos volúmenes o cargas de trabajo
3	1/2000	Baja tasa de fallas; similar al diseño anterior que, en el pasado, tenía bajas tasas de fallas para ciertos volúmenes o cargas de trabajo
4	1/1000	Tasa de fallas ocasional; similar al diseño anterior que, en el pasado, tenía bajas tasas de fallas para ciertos volúmenes o cargas de trabajo
5	1/500	Tasa de fallas moderada; similar al diseño anterior que, en el pasado, tenía moderadas tasas de fallas para ciertos volúmenes o cargas de trabajo
6	1/200	Tasa de fallas moderada a elevada; similar al diseño anterior que, en el pasado, tenía moderadas tasas de fallas para ciertos volúmenes o cargas de trabajo
7	1/100	Tasa de fallas alta; similar al diseño anterior que, en el pasado, tenía altas tasas de fallas que causaron problemas
8	1/50	Tasa de fallas alta; similar al diseño anterior que, en el pasado, tenía altas tasas de fallas que causaron problemas
9	1/20	Tasa de fallas muy alta; casi cierto causar problemas
10	1/10+	Tasa de fallas muy alta; casi cierto causar problemas

En esta presente sección, serán discutidos algunos aspectos básicos respecto a las distribuciones de probabilidades, también como serán presentadas algunas distribuciones útiles para el modelaje de un fenómeno en estudio. No se pretende agotar el asunto de las distribuciones, una vez que hay muchas distribuciones posibles ya bien conocidas en el campo de las estadísticas, permitiendo al lector profundizar en el asunto cuando necesite escoger una distribución de probabilidad que mejor se adecue al proceso de su interés.

Las distribuciones de probabilidad pueden ser definidas como discretas o como continuas.

Distribuciones discretas son aquellas funciones que poseen una variable aleatoria X , tal que puede asumir posibles valores finitos o infinitos numerables. Esos valores pueden ser puestos en una lista finita (x_1, x_2, \dots, x_n) o infinita contable (x_1, x_2, \dots)

Las distribuciones continuas poseen la variable aleatoria X que asume todos los posibles valores en un rango, por ejemplo, $0 \leq x \leq 1 \forall x \in \mathfrak{R}$.

Si se trata de una distribución discreta, la probabilidad de que x pueda asumir un determinado valor es $p(x)$.

$$P[X = x] = p(x) = p_x \quad (2.1)$$

En este caso, $p(x)$ es no-negativo para todo valor real de x . Y la suma de $p(x)$ para todos los posibles valores de x es 1, o sea:

$$\sum_j p_j = 1 \Rightarrow 0 \leq p(x) \leq 1 \quad (2.2)$$

En el caso de la distribución continua, la probabilidad de que x esté entre dos valores a y b es:

$$p[a \leq x \leq b] = \int_a^b f(x) dx \quad (2.3)$$

En el caso de la Ec. (2.3), para todo el intervalo Real, se queda:

$$\int_{-\infty}^{\infty} f(x) dx = 1 \quad (2.4)$$

Algunas funciones en Python hacen referencia a la probabilidad de masa y a la densidad de probabilidad. Cuando se refiera a la probabilidad de masa, se comprende que es una función de probabilidad discreta, mientras que la densidad de probabilidad se refiere a funciones de probabilidad continua.

La probabilidad asumida por una variable x es llamada función densidad de probabilidad (pdf).

El Código 2.2 presentará una gráfica (Fig. 2.1) de una distribución normal, con media $\mu = 0$ – lo que coincide con el parámetro de localización `loc=0` – y desviación estándar $\sigma = 1$ – lo que coincide con el parámetro de escala `scale=1` del método `norm.pdf()`, constante en la biblioteca

Scipy. También fueron usados los paquetes Numpy, para la computación científica, y Matplotlib, cuyo paquete Pyplot permite la generación de gráficas a partir de las rutinas desarrolladas con el Numpy.

El método “linspace” de la biblioteca Numpy crea un vector desde un valor inicial hasta un valor final, discretizado en n partes iguales (en este caso, $n = 100$).

```
# -*- coding: utf-8 -*-
from scipy.stats import norm
import matplotlib.pyplot as plt
import numpy as np
# %%
x = np.linspace(norm.ppf(0.01), norm.ppf(0.99), 100)
plt.plot(x, norm.pdf(x))
plt.title("Distribucion Normal (pdf)")
plt.xlabel("x")
plt.ylabel("Densidad de Probabilidad")
plt.show()
```

Código 2.1: Distribución Normal (pdf)

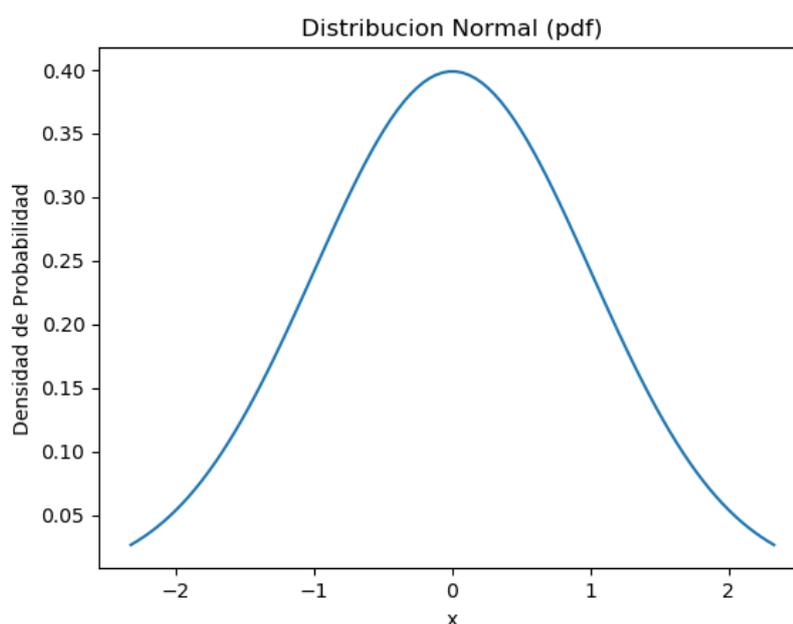


Figura 2.1. Función densidad de probabilidad normal (pdf)

La función distribución acumulativa (cdf) representa la probabilidad de que la variable asuma un valor menor o igual a x .

$$F(x) = P[X \leq x] = \alpha \quad (2.5)$$

Para una distribución continua, la cdf asume la forma matemática presente en la Ec. (2.6).

$$F(x) = \int_{-\infty}^x f(\mu) d\mu \quad (2.6)$$

Para una distribución discreta, puede ser escrita en acuerdo con la Ec. (2.7).

$$F(x) = \sum_{i=0}^x f(i) \quad (2.7)$$

El Código 2.2 presentará una gráfica (Fig. 2.2) con la cdf de una distribución normal, con los mismos parámetros del Código 4.1. Es importante percibir que el Código 4.2 tiene la misma estructura del Código 2.2, solamente cambiando el método para `norm.cdf()`. En la Fig. 2.2, se debe observar que el eje de las ordenadas fue nombrado como “Probabilidad”, mientras que en la Fig. 2.1 recibió el rótulo de “Densidad de Probabilidad”.

```
# -*- coding: utf-8 -*- 1
from scipy.stats import norm 2
import matplotlib.pyplot as plt 3
import numpy as np 4
# %% 5
x = np.linspace(norm.ppf(0.01), norm.ppf(0.99), 100) 6
plt.plot(x, norm.cdf(x)) 7
plt.title("Distribucion Normal (cdf)") 8
plt.xlabel("x") 9
plt.ylabel("Probabilidad") 10
plt.show() 11
```

Código 2.2: Distribución Normal (cdf)

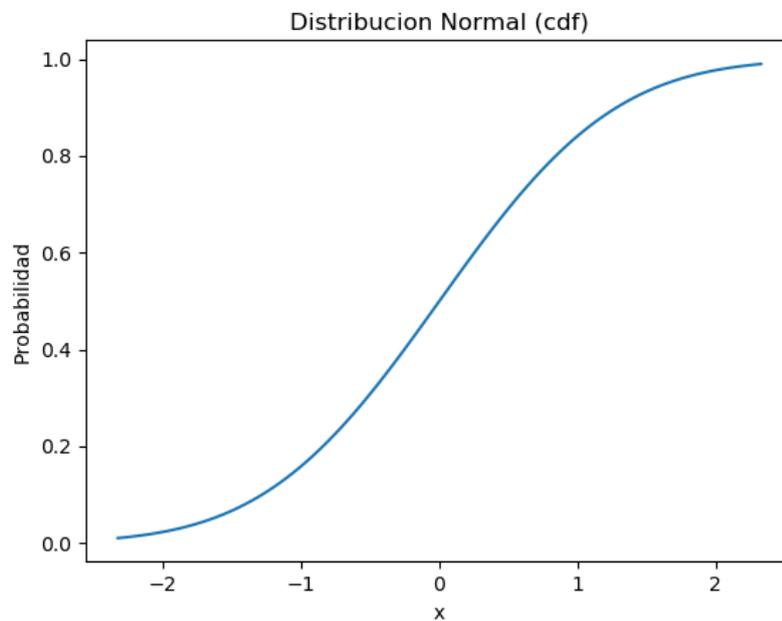


Figura 2.2. Función densidad acumulativa normal (cdf)

La función de punto porcentual (ppf), que es una función inversa de la cdf, permite calcular la probabilidad de que la variable sea menor o igual a x para un dado x . Es importante percibir en la Fig. 2.3 que hubo un cambio entre el eje de las ordenadas y el eje de las abscisas, en comparación con la Fig. 2.2. Para eso, el Código 2.2 tuvo que ser adaptado para generar el vector de valores de x , con el método `linspace()` de la biblioteca Numpy.

$$P[X \leq G(\alpha)] = \alpha \Rightarrow x = G(\alpha) = G(F(x)) \quad (2.8)$$

```

# -*- coding: utf-8 -*-
from scipy.stats import norm
import matplotlib.pyplot as plt
import numpy as np
# %%
x = np.linspace(0,1,100)
plt.plot(x, norm.ppf(x))
plt.title("Distribucion Normal (ppf)")
plt.xlabel("Probabilidad")
plt.ylabel("x")
plt.show()

```

Código 2.3: Distribución Normal (ppf)

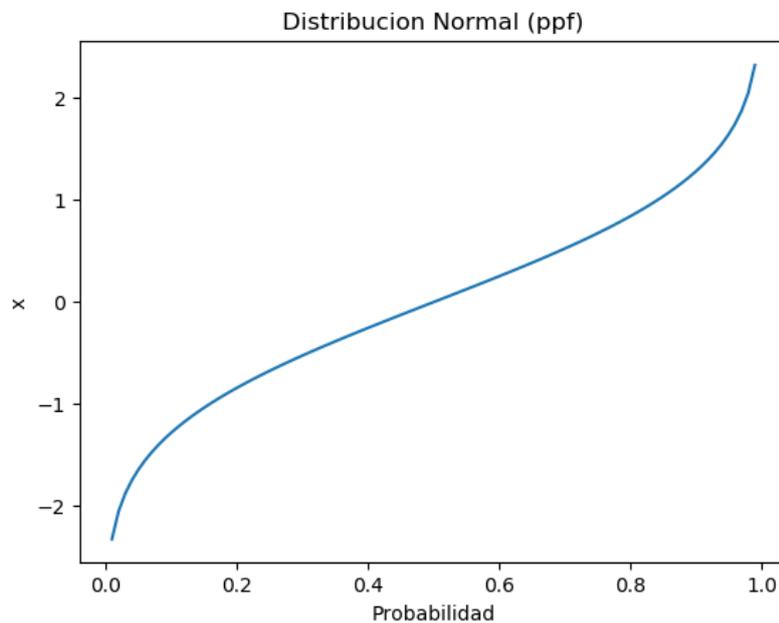


Figura 2.3. Función punto porcentual normal (ppf)

La función confiabilidad (R), también llamada de función de supervivencia, permite calcular la probabilidad de que la variable asuma un valor más grande que x .

$$R(x) = P[X > x] = 1 - F(x) \quad (2.9)$$

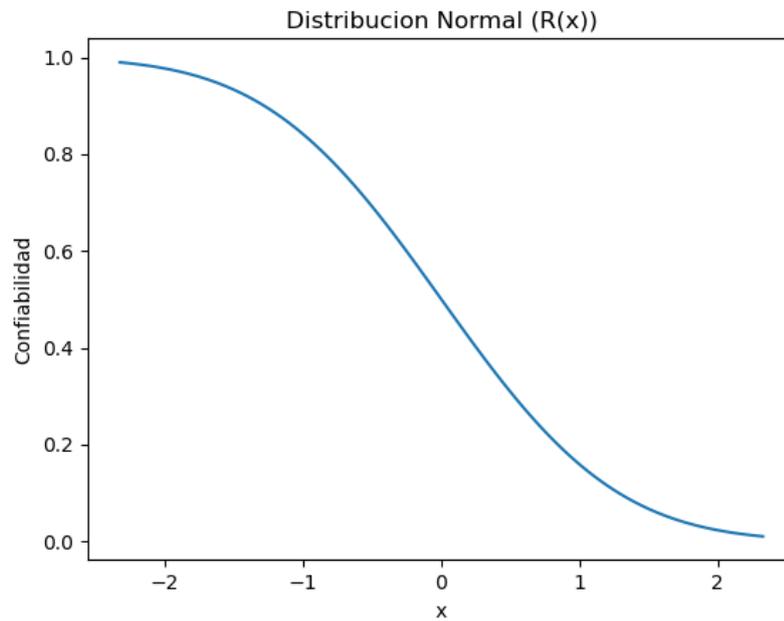


Figura 2.4. Función Confiabilidad (R)

```

# -*- coding: utf-8 -*-
from scipy.stats import norm
import matplotlib.pyplot as plt
import numpy as np
# %%
x = np.linspace(norm.ppf(0.01), norm.ppf(0.99), 100)
plt.plot(x, 1-norm.cdf(x))
plt.title("Distribucion Normal (R(x))")
plt.xlabel("x")
plt.ylabel("Confiabilidad")
plt.show()

```

Código 2.4: Distribución Normal (R)

La función de riesgo (h), también conocida en estudios de confiabilidad como tasa de fallas, es una razón entre la pdf y la función confiabilidad (R).

$$h(x) = \frac{f(x)}{R(x)} = \frac{f(x)}{1-F(x)} \quad (2.10)$$

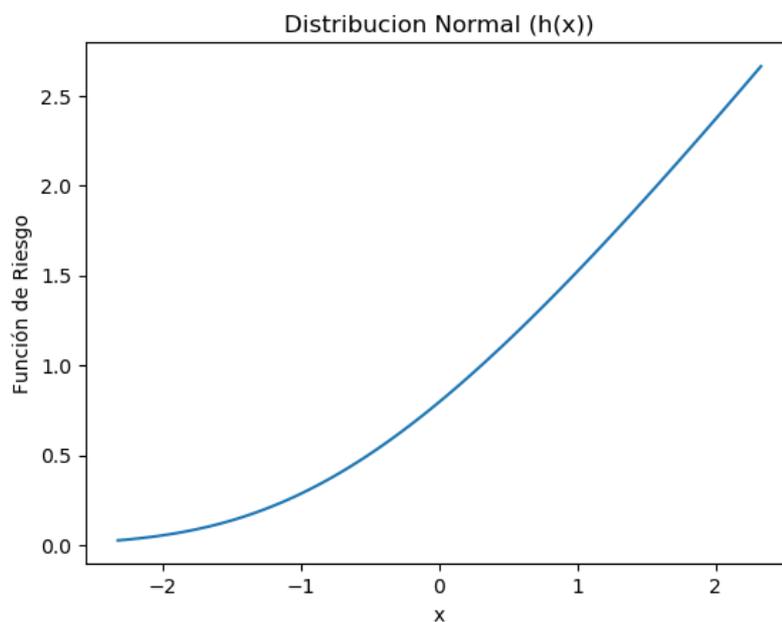


Figura 2.5. Función de Riesgo (h)

```

# -*- coding: utf-8 -*-      1
from scipy.stats import norm  2
import matplotlib.pyplot as plt  3
import numpy as np          4
# %%                          5
x = np.linspace(norm.ppf(0.01), norm.ppf(0.99), 100)  6
plt.plot(x, norm.pdf(x)/(1-norm.cdf(x)))  7
plt.title("Distribucion Normal (h(x))")  8
plt.xlabel("x")  9
plt.ylabel("Funcion de Riesgo")  10
plt.show()  11

```

Código 2.5: Distribución Normal (h)

La función de riesgo acumulativa (H), también conocida como tasa de fallas acumulativa, es definida como la integral de la función de riesgo (h).

$$H(x) = \int_{-\infty}^x h(\mu) d\mu \Rightarrow H(x) = -\ln(1 - F(x)) \quad (2.11)$$

```

# -*- coding: utf-8 -*-
from scipy.stats import norm
import matplotlib.pyplot as plt
import numpy as np
# %%
x = np.linspace(norm.ppf(0.01), norm.ppf(0.99), 100)
plt.plot(x, -np.log((1 - norm.cdf(x))))
plt.title("Distribucion Normal (H(x))")
plt.xlabel("x")
plt.ylabel("Funcion de Riesgo Cumulativa")
plt.show()

```

Código 2.6: Distribución Normal (H)

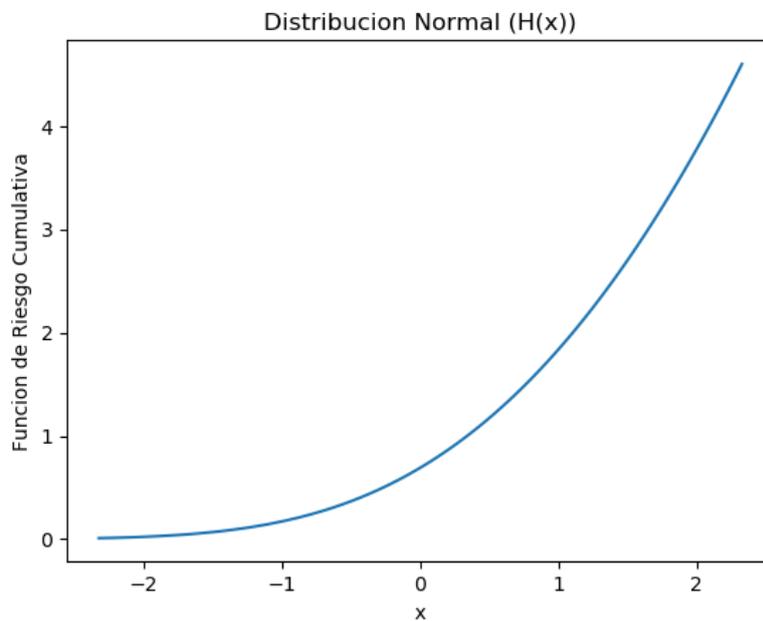


Figura 2.6. Función de Riesgo Cumulativa (H)

2.2.1 Parámetros de localización y de escala

En una distribución de probabilidad, el parámetro de localización está asociado a la posición de la gráfica de la pdf en el eje de las abscisas, mientras que el parámetro de escala está asociado a una mayor o menor compresión de la gráfica de la pdf alrededor de su valor de localización.

Como está citado en la Sección 2.2, en una distribución normal, el parámetro de localización es la media (μ) y el parámetro de escala es la desviación estándar (σ). La Fig. 2.7 ilustrará el comportamiento de las distribuciones para diferentes combinaciones de parámetros. Observe que, debido a que el área abajo de las curvas tiene valor unitario (en este caso, la gráfica es dicha normalizada), para dos curvas con el mismo valor medio, la que tiene mayor desviación estándar se presenta más achatada.

```

# -*- coding: utf-8 -*-
from scipy.stats import norm
import matplotlib.pyplot as plt
import numpy as np
#%%
x = np.linspace(norm.ppf(0.01), norm.ppf(0.99), 100)
plt.plot(x, norm.pdf(x, loc=0, scale=1), label='$\mu=0$ y $\sigma=1$')
plt.plot(x, norm.pdf(x, loc=0.5, scale=1), label='$\mu=0.5$ y $\sigma=1$')
plt.plot(x, norm.pdf(x, loc=0, scale=1.5), label='$\mu=0$ y $\sigma=1.5$')
plt.title("Distribucion Normal (pdf)")
plt.xlabel("x")
plt.ylabel("Densidad de Probabilidad")
plt.legend(loc='best', frameon=False)
plt.show()

```

Código 2.7: Distribución Normal

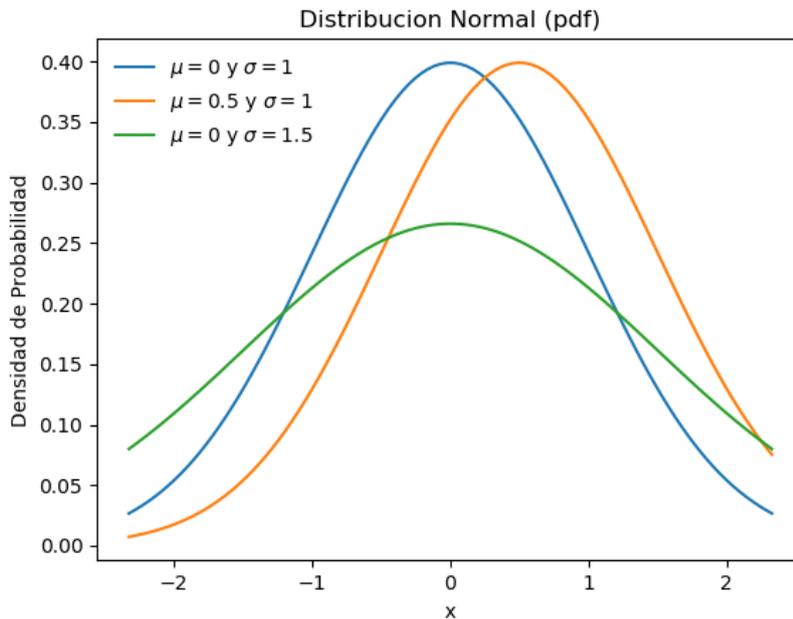


Figura 2.7. Distribuciones normales para 3 combinaciones de parámetros

Considerando que a y b son los valores de los parámetros de localización y de escala, respectivamente, y que el soporte es el intervalo cerrado $[0,1]$, las fórmulas de distribución se quedan como están a continuación:

- **Función densidad de probabilidad (pdf)** - $f(x, a, b) = \left(\frac{1}{b}\right)f\left(\frac{x-a}{b}\right)$;
- **Función distribución acumulativa (cdf)** - $F(x, a, b) = F\left(\frac{x-a}{b}\right)$;
- **Función punto porcentual (ppf)** - $G(\alpha, a, b) = a + bG(\alpha)$;
- **Función confiabilidad (R)** - $R(x, a, b) = R\left(\frac{x-a}{b}\right)$;
- **Función de riesgo (h)** - $h(x, a, b) = \left(\frac{1}{b}\right)h\left(\frac{x-a}{b}\right)$;
- **Función de riesgo acumulativa (H)** - $H(x, a, b) = H\left(\frac{x-a}{b}\right)$;
- **Números al azar de una distribución uniforme** - $Y(a, b) = a + bY(0, 1)$.

2.2.2 Principales estadísticas

En estudios en los cuales son empleados modelos matemáticos no deterministas (o aleatorios), algunos parámetros caracterizan las distribuciones y suministran una valiosa información respecto al sistema modelado. En esta sección, serán presentados algunos de esos parámetros, denominados *estadísticas* de las distribuciones.

Las estadísticas serán definidas de manera formal abajo, pero sus cálculos podrán ser automáticamente obtenidos por funciones ya disponibles en Python, sin la necesidad del usuario desarrollar todo el cálculo numérico involucrado con las ecuaciones matemáticas presentadas.

Definición 2.2.1 — Esperanza matemática. Es también conocida como esperanza, valor esperado, media. En un modelo de un sistema que se somete a valores al azar, la esperanza matemática está asociada al valor medio del fenómeno en estudio. Puede ser interpretada como el centro de gravedad de la unidad de masa sobre una recta [Meyer (1983)].

La esperanza \mathbb{E} de un conjunto de datos representados por una variable aleatoria X , discreta, es dada por la siguiente ecuación [Meyer (1983)]:

$$\mathbb{E}(X) = \sum_{i=1}^{\infty} x_i p(x_i) \quad (2.12)$$

Para los sistemas de modelos continuos, la esperanza se queda:

$$\mathbb{E}(X) = \int_{-\infty}^{\infty} y f(y) dy \quad (2.13)$$

Por ejemplo, para una distribución normal, la esperanza matemática es la media de la distribución, o sea, $\mathbb{E} = \mu$. Para una distribución de Weibull, la media es $\mathbb{E} = \Gamma(\frac{\gamma+1}{\gamma})$, donde Γ es la función gamma y γ es el parámetro de forma de la distribución, a ser discutido con más detalles en este texto.

Definición 2.2.2 — Mediana. Es una medida de localización en donde la mitad de los datos son más pequeños que su valor y la otra mitad está compuesta por valores más grandes.

La mediana de un conjunto de datos representados por una variable aleatoria X está dada por las siguientes ecuaciones [NIST/SEMATECH (2012)]:

Si N es par:

$$\tilde{x} = x_{(N+1)/2} \quad (2.14)$$

Si N es impar:

$$\tilde{x} = \frac{(x_{N/2} + x_{(N/2)+1})}{2} \quad (2.15)$$

Definición 2.2.3 — Moda. Es una información respecto a cuál valor de la variable aleatoria ocurre con más frecuencia.

Definición 2.2.4 — Varianza. Es una medida de dispersión que está asociada con el cuadrado de la desviación de la variable respecto a su valor medio. Puede ser interpretada como el momento de inercia de la unidad de masa (esperanza) sobre un eje perpendicular que pase por el centro de gravedad de la masa [Meyer (1983)].

La varianza $V(X) = \sigma^2$ de un conjunto de datos representados por una variable aleatoria X está dada por la siguiente ecuación [Meyer (1983)]:

$$V(X) = \mathbb{E}[X - \mathbb{E}(X)]^2 = \mathbb{E}(X^2) - [\mathbb{E}(X)]^2 \quad (2.16)$$

Definición 2.2.5 — Desviación estándar. Es una medida de dispersión de un conjunto de datos numéricos. Es la raíz cuadrada del valor de la varianza.

$$\sigma = \sqrt{V(X)} \quad (2.17)$$

2.2.3 Distribución normal

La distribución normal es una de las más conocidas distribuciones de probabilidad. Según [Stewart (2013)], cerca del año de 1730, Abraham de Moivre dedujo una fórmula aproximada para el cálculo de las probabilidades involucradas con lanzamientos repetidos de una moneda viciada. Esa fórmula llevó a la función error, conocida como distribución normal, muchas veces llamada de “curva de la campana”, debido a su formato.

Es una distribución importante porque, para una variable aleatoria X que pueda ser representada por la suma de cualquier n variables aleatorias independientes (que satisfagan a determinadas condiciones), entonces esa suma, para n suficientemente gran, tendrá distribución aproximadamente normal [Meyer (1983)]. O sea, por el Teorema del Límite Central, en una distribución con una cantidad muy gran de muestras, la distribución muestral de su media tenderá para una distribución normal, respecto la variable original. Además de eso, la desviación estándar de la distribución muestral de la media se aproxima a σ/\sqrt{n} .

Su función distribución de probabilidad está dada por la fórmula:

$$f(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}} \quad (2.18)$$

En la Ec. (2.18), μ y σ son la media y la desviación estándar, respectivamente.

Empleando la Ec. (2.18) en el Código 2.8, tendremos el mismo resultado obtenido en el Código 2.2, resultando en la misma Fig. 2.1, pero ahora usando el concepto del paradigma funcional de la programación.

```

# -*- coding: utf-8 -*-
from scipy.stats import norm
import matplotlib.pyplot as plt
import numpy as np
#%%
x = np.linspace(norm.ppf(0.01), norm.ppf(0.99), 100)
mu=0
sigma=1
def normal(x,med,de):
    y=(np.exp(-((x-med)**2)/(2*(de**2))))/(de*np.sqrt(2*np.pi))
    return y
plt.plot(x, normal(x,mu,sigma))
plt.title("Distribucion Normal (pdf)")
plt.xlabel("x")
plt.ylabel("Densidad de Probabilidad")
plt.show()

```

Código 2.8: Distribución Normal

En el paradigma funcional, la función en Python empieza por la definición, cuya palabra “def” pertenece al diccionario interno del lenguaje. A continuación, el usuario define un nombre cualquiera para la función (en ese caso, la palabra “normal”) y entre paréntesis pone las variables que serán procesadas por la función. Es importante percibir que la sintaxis del Python considera la tabulación como detalle fundamental para la interpretación del código. La salida de la función, en el caso del Código 2.8, fue suministrada por la instrucción después de la palabra interna “return”.

La cdf de la función normal es dada por la Ec. (2.19), donde “erf” es la función error (Ec. (2.20)).

$$F(x) = \frac{1}{2} \left(1 + \operatorname{erf} \frac{x-\mu}{\sigma\sqrt{2}} \right) \quad (2.19)$$

$$\operatorname{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^x e^{-t^2} dt = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (2.20)$$

La función error, en Python, puede ser llamada empleándose la biblioteca Scipy, cuyo paquete Special ya dispone de una rutina interna. El Código 2.9 presentará una solución que emplea el paradigma funcional, con funciones compuestas, al que se debe comparar con el Código 2.2 y percibir que el uso de las bibliotecas preestablecidas es mucho más práctico de implementarse. El método “quad” indica que la integración fue ejecutada por el método de la cuadratura de Gauss. El Código 2.9 resulta en la Fig. 2.2.

```

# -*- coding: utf-8 -*-      1
from scipy.stats import norm  2
from scipy.integrate import quad  3
import matplotlib.pyplot as plt  4
import numpy as np          5
#%%                          6
n=100 #numero de elementos en la discretizacion  7
x = np.linspace(norm.ppf(0.01), norm.ppf(0.99), n)  8
mu=0                          9
sigma=1                       10
z=np.zeros(n)                 11
def integ(x): #funcion en el integrando de la funcion error  12
    return np.exp(-(x**2))    13
def error(x):                 14
    y=2/np.sqrt(np.pi)*quad(integ,0,x)[0]  15
    return y                  16
def cdfnormal(x,med,de):     17
    for i in range(n):       18
        z[i]=1/2*(1+error((x[i]-med)/(de*np.sqrt(2))))  19
    return z                 20
plt.plot(x, cdfnormal(x,mu,sigma))  21
plt.title("Distribucion Normal (cdf)")  22
plt.xlabel("x")              23
plt.ylabel("Probabilidad")   24
plt.show()                   25

```

Código 2.9: Distribución Normal (cdf)

El usuario que posee los datos de la pdf y de la cdf, puede fácilmente obtener las demás funciones de interés para este libro, que son la ppf, la $R(x)$, la $h(x)$ y la $H(x)$, ya discutidas en la Sección 2.2.

2.2.4 Distribución uniforme

La distribución uniforme encuentra su uso tal vez más común en la generación de números al azar, igualmente distribuidos (es decir, con pesos iguales para sus elecciones), en un intervalo $a \leq x \leq b$.

$$f(x) = \frac{1}{b-a} \Rightarrow f(x) = 1 \quad \forall 0 \leq x \leq 1 \quad (2.21)$$

Tratándose de distribuciones de probabilidad, donde el área de las gráficas son normalizadas, como lo observado en la Ec. 2.21, el valor de la densidad de probabilidad es una constante y, en caso de un soporte en el rango $0 \leq x \leq 1$, su valor es necesariamente igual a la unidad. El Código 2.10 y la Fig. 2.8 ilustran ese hecho.

```
# -*- coding: utf-8 -*-      1
from scipy.stats import uniform      2
import matplotlib.pyplot as plt      3
import numpy as np              4
# %%                             5
x = np.linspace(0, 1, 100)          6
plt.plot(x, uniform.pdf(x))         7
plt.title("Distribucion Uniforme (pdf)")  8
plt.xlabel("x")                     9
plt.ylabel("Densidad de Probabilidad") 10
plt.show()                          11
```

Código 2.10: Distribución Uniforme (pdf)

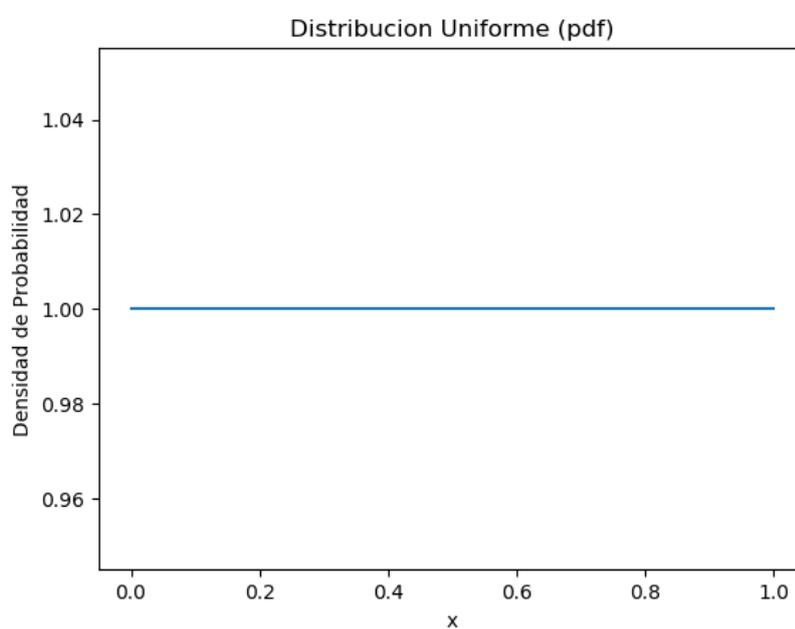


Figura 2.8. Distribución uniforme estándar

La cdf es la integral de la pdf. En el caso de la distribución uniforme estándar, la cdf se queda como en la Fig. 2.9, en el intervalo $0 \leq x \leq 1$.

$$F(x) = x \quad \forall 0 \leq x \leq 1 \quad (2.22)$$

```

# -*- coding: utf-8 -*-
from scipy.stats import uniform
import matplotlib.pyplot as plt
import numpy as np
# %%
x = np.linspace(0, 1, 100)
plt.plot(x, uniform.cdf(x))
plt.title("Distribucion Uniforme (cdf)")
plt.xlabel("x")
plt.ylabel("Probabilidad")
plt.show()

```

Código 2.11: Distribución Uniforme Estándar (cdf)

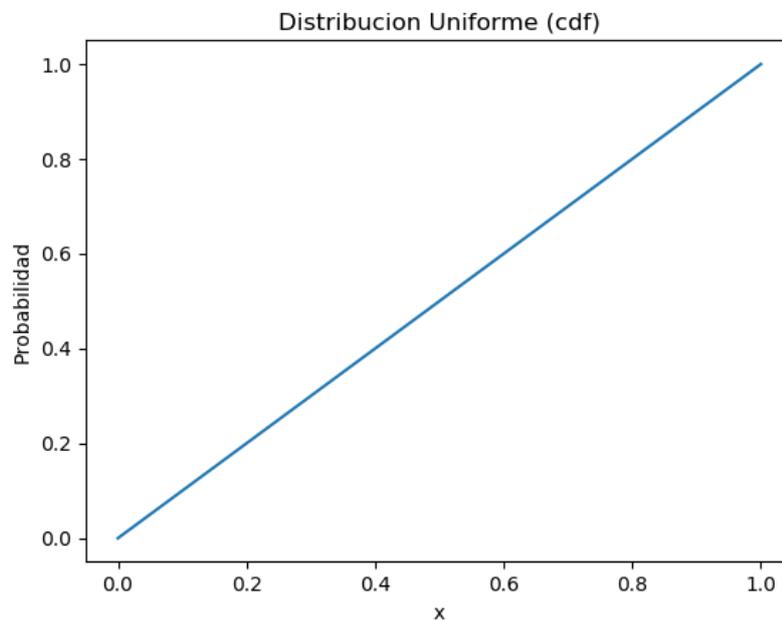


Figura 2.9. Distribución uniforme estándar

2.2.5 Distribución de Weibull

La distribución de Weibull, desarrollada por el matemático y ingeniero sueco Ernst Hjalmar Waloddi Weibull (1887-1979), recibe especial atención porque, al depender de los valores de sus parámetros, su modelo puede adaptarse a una gran variedad de comportamientos físicos de los sistemas. Por ejemplo, en los casos en que la tasa de fallas de un sistema está asociada a la duración de la vida de una pieza – es el caso de una ley de fallas exponencial – los parámetros de la distribución de Weibull pueden ser ajustados para representar la probabilidad de fallas del sistema que contiene esa pieza.

La pdf de una distribución de Weibull está dada por la Ec. (2.23), donde γ es el parámetro de forma:

$$f(x) = \frac{\gamma}{b} \left(\frac{x-a}{b}\right)^{\gamma-1} e^{-((x-a)/b)^\gamma} \quad \forall x \geq a; \gamma, b > 0 \quad (2.23)$$

El Código 2.12 presentará el comportamiento de 4 combinaciones de parámetros de la distribución de Weibull, donde varía el valor de γ y se mantienen constantes los valores de $a = 0$ y $b = 1$, en la Ec. 2.23.

```

# -*- coding: utf-8 -*-      1
from scipy.stats import weibull_min      2
import matplotlib.pyplot as plt      3
import numpy as np      4
# %%      5
x = np.linspace(0, 5, 100)      6
c1=0.5      7
c2=1      8
c3=2      9
c4=5      10
      11
y1 = weibull_min.pdf(x, c1)      12
y2 = weibull_min.pdf(x, c2)      13
y3 = weibull_min.pdf(x, c3)      14
y4 = weibull_min.pdf(x, c4)      15
      16
plt.subplot(2, 2, 1)      17
plt.title("Distribucion de Weibull ( $\gamma$ =%0.1f)" %c1)      18
plt.plot(x, y1)      19
plt.ylabel("Densidad de probabilidad")      20
plt.gca().axes.get_xaxis().set_visible(False)      21
      22
plt.subplot(2, 2, 2)      23
plt.title("Distribucion de Weibull ( $\gamma$ =%0.1f)" %c2)      24
plt.plot(x, y2)      25
plt.gca().axes.get_xaxis().set_visible(False)      26
      27
plt.subplot(2, 2, 3)      28
plt.title("Distribucion de Weibull ( $\gamma$ =%0.1f)" %c3)      29
plt.plot(x, y3)      30
plt.xlabel("x")      31
plt.ylabel("Densidad de probabilidad")      32
      33
plt.subplot(2, 2, 4)      34
plt.title("Distribucion de Weibull ( $\gamma$ =%0.1f)" %c4)      35
plt.plot(x, y4)      36
plt.xlabel("x")      37
      38
plt.show()      39

```

Código 2.12: Distribución de Weibull Estándar (pdf)

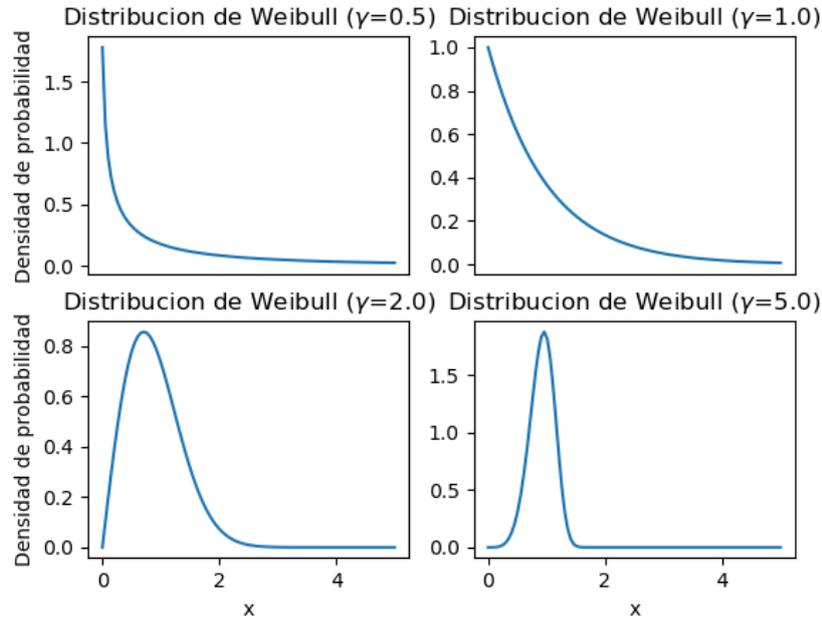


Figura 2.10. Distribución de Weibull estándar

Las siguientes ecuaciones presentarán los principales modelos de utilidad para el análisis del comportamiento de un sistema sometido a condiciones que se ajustan bien a una distribución de Weibull. En estos casos, se mantienen constantes los valores de $a = 0$ y $b = 1$.

- **Función distribución acumulativa (cdf)** - $F(x) = 1 - e^{-(x^\gamma)} \forall x \geq 0; \gamma > 0$;
- **Función punto porcentual (ppf)** - $G(\alpha) = (-\ln(1 - \alpha))^{1/\gamma} \forall 0 \leq \alpha < 1; \gamma > 0$;
- **Función confiabilidad (R)** - $R(x) = e^{-(x^\gamma)} \forall x \geq 0; \gamma > 0$;
- **Función de riesgo (h)** - $h(x) = \gamma x^{\gamma-1} \forall x \geq 0; \gamma > 0$;
- **Función de riesgo acumulativa (H)** - $H(x) = x^\gamma \forall x \geq 0; \gamma > 0$;
- **Media** - $\Gamma\left(\frac{\gamma+1}{\gamma}\right)$;
- **Mediana** - $\ln(2)^{1/\gamma}$;
- **Moda** - $\left(1 - \frac{1}{\gamma}\right)^{1/\gamma} \forall \gamma > 1; 0 \forall \gamma \leq 1$;
- **Desviación estándar** - $\sqrt{\Gamma\left(\frac{\gamma+2}{\gamma}\right) - \left(\Gamma\left(\frac{\gamma+1}{\gamma}\right)\right)^2}$;

2.2.6 Distribución beta

La distribución beta tiene gran aplicabilidad en el modelaje de fallas, tanto por su versatilidad – por la combinación de valores de sus parámetros – cuanto por permitir modelar un tipo especial de falla relacionada con la edad del equipo. Esa falla, presentada en la Fig. 1.1 es denominada “curva de la bañera” (Fig. 3.3) y será explorada oportunamente en este libro.

La pdf de la distribución beta es presentada en la Ec. (2.24), con parámetros p y q denominados parámetros de forma, y parámetros a y b denominados límites inferior y superior, respectivamente. Cuando $a = 0$ y $b = 1$, la distribución asume su forma estándar.

$$f(x) = \frac{(x-a)^{p-1}(b-x)^{q-1}}{B(p,q)(b-a)^{p+q-1}} \forall a \leq x \leq b; p, q > 0 \quad (2.24)$$

La función beta puede ser calculada por la Ec. (2.25):

$$B(\alpha, \beta) = \int_0^1 t^{\alpha-1} (1-t)^{\beta-1} dt \quad (2.25)$$

Su parámetro de localización es a y de escala es $b - a$.

El Código 2.13 y la Fig. 2.11 ilustran 4 combinaciones de parámetros de la distribución beta.

```

# -*- coding: utf-8 -*-
from scipy.integrate import quad
import matplotlib.pyplot as plt
import numpy as np
# %%
x = np.linspace(0, 1, 100)
p1=0.5
p2=2
q1=0.5
q2=2
a=0
b=1
# %%
def integ(x, a, b):
    return x**(a-1)*(1-x)**(b-1)
def beta(p, q):
    return quad(integ, 0, 1, args=(a, b))[0]
def pdf(x, a, b, p, q):
    return ((x-a)**(p-1)*(b-x)**(q-1))/(beta(p, q)*(b-a)**(p+q-1))

y1 = pdf(x, a, b, p1, q1)
y2 = pdf(x, a, b, p1, q2)
y3 = pdf(x, a, b, p2, q1)
y4 = pdf(x, a, b, p2, q2)

plt.subplot(2, 2, 1)
plt.title("(p=%0.1f, q=%0.1f)" % (p1, q1))
plt.plot(x, y1)
plt.ylabel("Densidad de probabilidad")
plt.gca().axes.get_xaxis().set_visible(False)

plt.subplot(2, 2, 2)
plt.title("(p=%0.1f, q=%0.1f)" % (p1, q2))
plt.plot(x, y2)
plt.gca().axes.get_xaxis().set_visible(False)

plt.subplot(2, 2, 3)
plt.title("(p=%0.1f, q=%0.1f)" % (p2, q1))
plt.plot(x, y3)
plt.xlabel("x")
plt.ylabel("Densidad de probabilidad")

plt.subplot(2, 2, 4)
plt.title("(p=%0.1f, q=%0.1f)" % (p2, q2))
plt.plot(x, y4)
plt.xlabel("x")

plt.show()

```

Código 2.13: Distribución beta estándar (pdf)

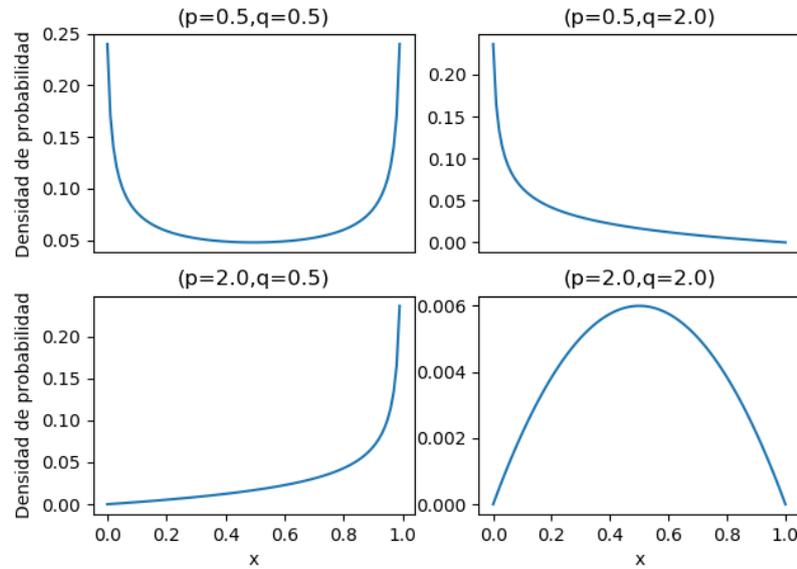


Figura 2.11. Distribución beta estándar

Las siguientes ecuaciones presentarán los principales modelos de utilidad para el análisis del comportamiento de un sistema sometido a condiciones que se ajustan bien a una distribución beta. En estos casos, se mantienen constantes los valores de $a = 0$ y $b = 1$.

- **Función distribución acumulativa (cdf)** - $F(x) = I_x(p, q) = \frac{\int_0^x t^{p-1}(1-t)^{q-1} dt}{B(p, q)} \quad \forall 0 \leq x \leq 1; p, q > 0$;
- **Función punto porcentual (ppf)** - la ppf no posee una forma cerrada. Su cálculo deberá ser ejecutado por computación numérica [NIST/SEMATECH (2012)];
- **Media** - $\frac{p}{p+q}$;
- **Moda** - $\frac{p-1}{p+q-2} \quad \forall p, q \geq 1$;
- **Desviación estándar** - $\sqrt{\frac{pq}{(p+q)^2(p+q+1)}}$;

2.2.7 Distribución binomial

Este es un tipo de distribución discreta, donde se hacen estimaciones a partir de eventos mutuamente excluyentes, del tipo “éxito” o “fracaso”.

A partir de la observación de x éxitos en N tentativas, la probabilidad de éxito es denominada p . Asumiéndose que p es un valor fijo para todas las tentativas, su función de probabilidad de masa se queda como la Ec. (2.26):

$$P(x, p, n) = \binom{n}{x} p^x (1-p)^{n-x} \quad \forall x = 0, 1, 2, \dots, n \quad (2.26)$$

$$\binom{n}{x} = \frac{n!}{x!(n-x)!} \quad (2.27)$$

El Código 2.14 y la Fig. 2.12 ilustran 4 combinaciones de parámetros de la distribución binomial. Observe que fue empleado el método `arange()`, de la biblioteca Numpy, para generar el vector de las abscisas. Ese método divide un intervalo en partes iguales.

```
# -*- coding: utf-8 -*-      1
from scipy.stats import binom      2
import matplotlib.pyplot as plt      3
import numpy as np      4
#%%      5
p1=0.1      6
p2=0.25      7
p3=0.5      8
p4=0.75      9
n=100      10
x = np.arange(0,100)      11
#%%      12
y1 = binom.pmf(x, n, p1)      13
y2 = binom.pmf(x, n, p2)      14
y3 = binom.pmf(x, n, p3)      15
y4 = binom.pmf(x, n, p4)      16

plt.subplot(2, 2, 1)      17
plt.title("Binomial (p=%0.2f,n=%d)" %(p1,n))      18
plt.plot(x, y1)      19
plt.ylabel("Probabilidad de masa")      20
plt.gca().axes.get_xaxis().set_visible(False)      21
      22
plt.subplot(2, 2, 2)      23
plt.title("Binomial (p=%0.2f,n=%d)" %(p2,n))      24
plt.plot(x, y2)      25
plt.gca().axes.get_xaxis().set_visible(False)      26
      27
plt.subplot(2, 2, 3)      28
plt.title("Binomial (p=%0.2f,n=%d)" %(p3,n))      29
plt.plot(x, y3)      30
plt.xlabel("x")      31
plt.ylabel("Probabilidad de masa")      32
      33
plt.subplot(2, 2, 4)      34
plt.title("Binomial (p=%0.2f,n=%d)" %(p4,n))      35
plt.plot(x, y4)      36
plt.xlabel("x")      37
      38
plt.show()      39
      40
```

Código 2.14: Distribución binomial (pmf)

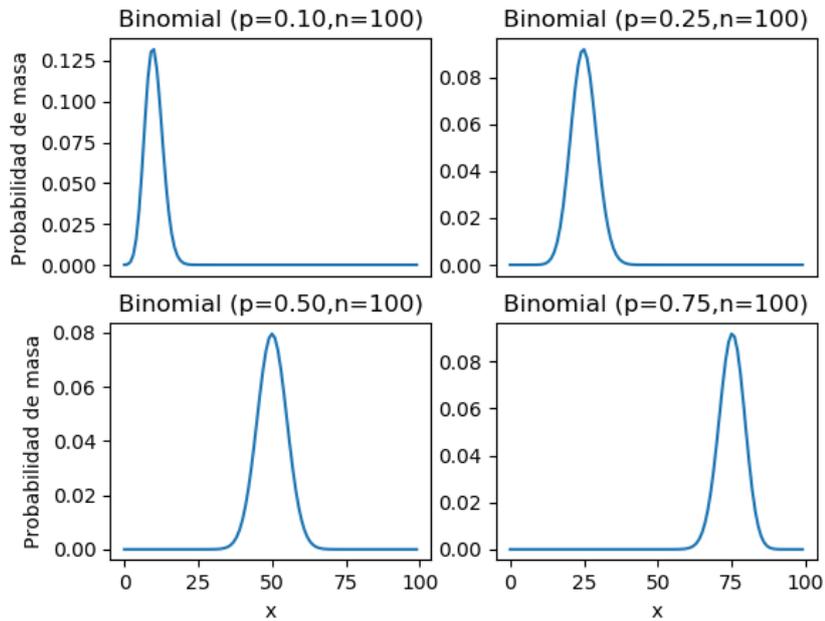


Figura 2.12. Distribución binomial

Las siguientes ecuaciones presentarán los principales modelos de utilidad para el análisis del comportamiento de un sistema sometido a condiciones que se ajustan bien a una distribución binomial.

- **Función distribución acumulativa (cdf)** - $F(x, p, n) = \sum_{i=0}^x \binom{n}{i} p^i (1-p)^{n-i}$;
- **Función punto porcentual (ppf)** - la ppf no posee una forma cerrada. Su cálculo deberá ser ejecutado por computación numérica [NIST/SEMATECH (2012)];
- **Media** - np ;
- **Moda** - $p(n+1) \forall 1 \leq x \leq p(n+1)$;
- **Desviación estándar** - $\sqrt{np(1-p)}$.

El Código 2.15 y la Fig. 2.13 ilustran los resultados para la generación de la cdf de la distribución binomial ilustrada por la Fig. 2.12.

```
# -*- coding: utf-8 -*-
from scipy.stats import binom
import matplotlib.pyplot as plt
import numpy as np
#%%
p1=0.1
p2=0.25
p3=0.5
p4=0.75
n=100
x = np.arange(0,100)
#%%
y1 = binom.cdf(x, n, p1)
y2 = binom.cdf(x, n, p2)
y3 = binom.cdf(x, n, p3)
y4 = binom.cdf(x, n, p4)

plt.subplot(2, 2, 1)
plt.title("Binomial cdf (p=%0.2f,n=%d)" %(p1,n))
plt.plot(x, y1)
plt.ylabel("Probabilidad")
plt.gca().axes.get_xaxis().set_visible(False)

plt.subplot(2, 2, 2)
plt.title("Binomial cdf (p=%0.2f,n=%d)" %(p2,n))
plt.plot(x, y2)
plt.gca().axes.get_xaxis().set_visible(False)

plt.subplot(2, 2, 3)
plt.title("Binomial cdf (p=%0.2f,n=%d)" %(p3,n))
plt.plot(x, y3)
plt.xlabel("x")
plt.ylabel("Probabilidad")

plt.subplot(2, 2, 4)
plt.title("Binomial cdf (p=%0.2f,n=%d)" %(p4,n))
plt.plot(x, y4)
plt.xlabel("x")

plt.show()
```

Código 2.15: Distribución binomial (cdf)

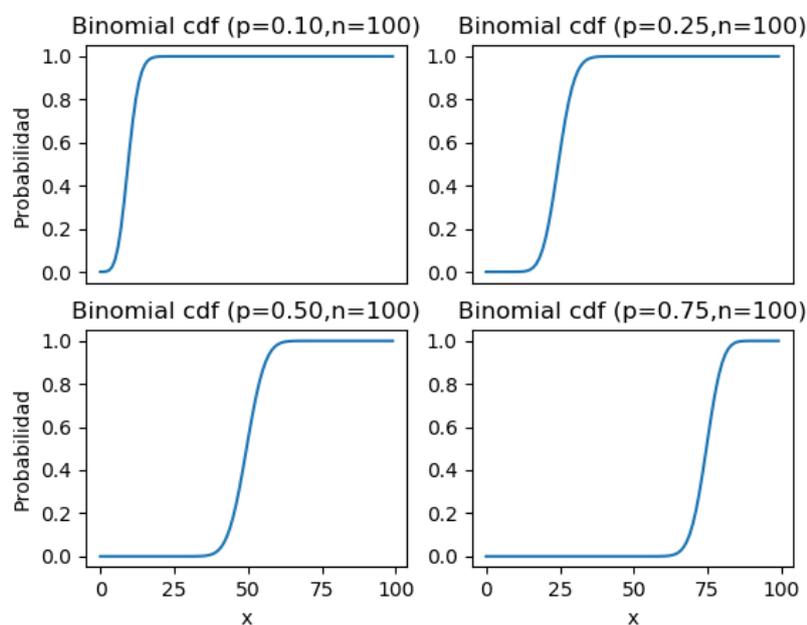


Figura 2.13. Distribución binomial (cdf)

2.2.8 Distribución de Poisson

La distribución de Poisson es especialmente útil para el cálculo de tasas de fallas (λ) de sistemas, cuando esas fallas ocurren en media a intervalos regulares.

Su función de probabilidad de masa se queda como la Ec. (2.28):

$$P(x, \lambda) = \frac{e^{-\lambda} \lambda^x}{x!} \quad \forall x = 0, 1, 2, \dots \quad (2.28)$$

El Código 2.16 y la Fig. 2.14 ilustran 4 combinaciones de parámetros de la distribución binomial. Observe que no se trata esa distribución como pdf, pero sí como pmf (probabilidad de masa).

```
# -*- coding: utf-8 -*- 1
from scipy.stats import poisson 2
import matplotlib.pyplot as plt 3
import numpy as np 4
#%% 5
l1=5 6
l2=15 7
l3=25 8
l4=35 9
x = np.arange(0,50) 10
#%% 11
y1 = poisson.pmf(x, l1) 12
y2 = poisson.pmf(x, l2) 13
y3 = poisson.pmf(x, l3) 14
y4 = poisson.pmf(x, l4) 15
16
plt.subplot(2, 2, 1) 17
plt.title("Poisson pmf ( $\lambda$ =%d)" %l1) 18
plt.plot(x, y1) 19
plt.ylabel("Probabilidad") 20
plt.gca().axes.get_xaxis().set_visible(False) 21
22
plt.subplot(2, 2, 2) 23
plt.title("Poisson pmf ( $\lambda$ =%d)" %l2) 24
plt.plot(x, y2) 25
plt.gca().axes.get_xaxis().set_visible(False) 26
27
plt.subplot(2, 2, 3) 28
plt.title("Poisson pmf ( $\lambda$ =%d)" %l3) 29
plt.plot(x, y3) 30
plt.xlabel("x") 31
plt.ylabel("Probabilidad") 32
33
plt.subplot(2, 2, 4) 34
plt.title("Poisson pmf ( $\lambda$ =%d)" %l4) 35
plt.plot(x, y4) 36
plt.xlabel("x") 37
38
plt.show() 39
```

Código 2.16: Distribución de Poisson (pmf)

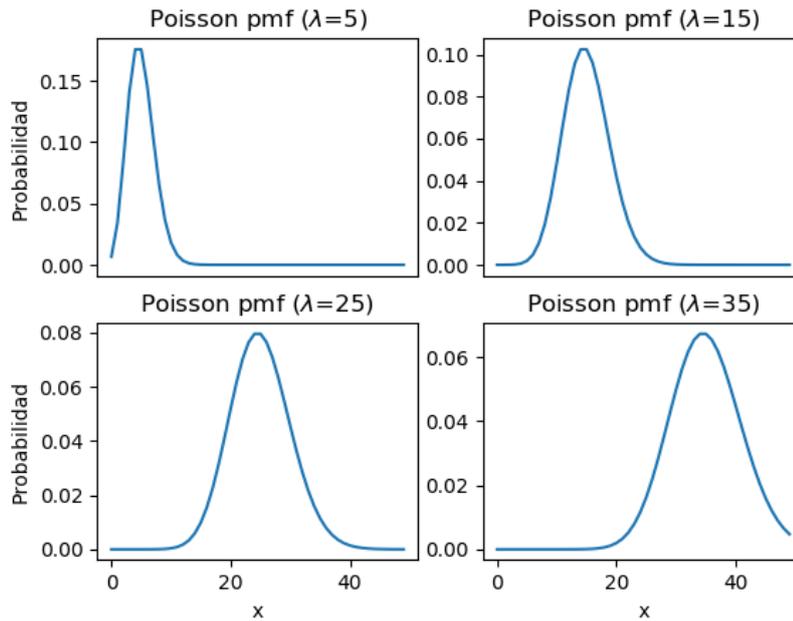


Figura 2.14. Distribución de Poisson (pmf)

Las siguientes ecuaciones presentarán los principales modelos de utilidad para el análisis del comportamiento de un sistema sometido a condiciones que se ajustan bien a una distribución de Poisson.

- **Función distribución acumulativa (cdf)** - $F(x, \lambda) = \sum_{i=0}^x \frac{e^{-\lambda} \lambda^i}{i!}$;
- **Función punto porcentual (ppf)** - la ppf no posee una forma cerrada. Su cálculo deberá ser ejecutado por computación numérica [NIST/SEMATECH (2012)];
- **Media** - λ ;
- **Moda** - para números no enteros, es el número más grande que sea menor que λ . Para un entero λ , $x = \lambda$ y $x = \lambda - 1$ son ambos la moda.
- **Desviación estándar** - $\sqrt{\lambda}$.

2.3 Estimación de parámetros y ajuste de distribuciones

Una vez que se recolecten los datos numéricos, es necesario transformar esos datos en distribuciones de probabilidades para que el análisis de esas distribuciones permitan al analista llegar a conclusiones y hacer previsiones.

Para eso, es importante ajustar una distribución a los datos obtenidos, de manera que la distribución permita los mejores resultados posibles. La naturaleza nos está obligada a seguir ninguna distribución específica, pero la gran cantidad de distribuciones de probabilidad permite hacer ajustes óptimos, de manera que las previsiones sean las mejores posibles. Los datos no son deterministas, pero las previsiones pueden hacerse bien justas y precisas a depender de la calidad y de la cantidad de esos datos.

Hay muchas posibles distribuciones estadísticas que se adecuan a los datos experimentales. Entre distribuciones de características muy parecidas, hay técnicas que permiten al analista decidir cuál de esas distribuciones es mejor que la otra.

Por ejemplo, con los datos generados al azar y puestos en el histograma de la Fig. 2.15 se ajustarán 3 posibles distribuciones. Muchas otras distribuciones de probabilidades podrían ser candidatas (vea [NIST/SEMATECH (2012)] y la biblioteca de funciones estadísticas Scipy Stats), pero ese es un mero ejemplo didáctico.

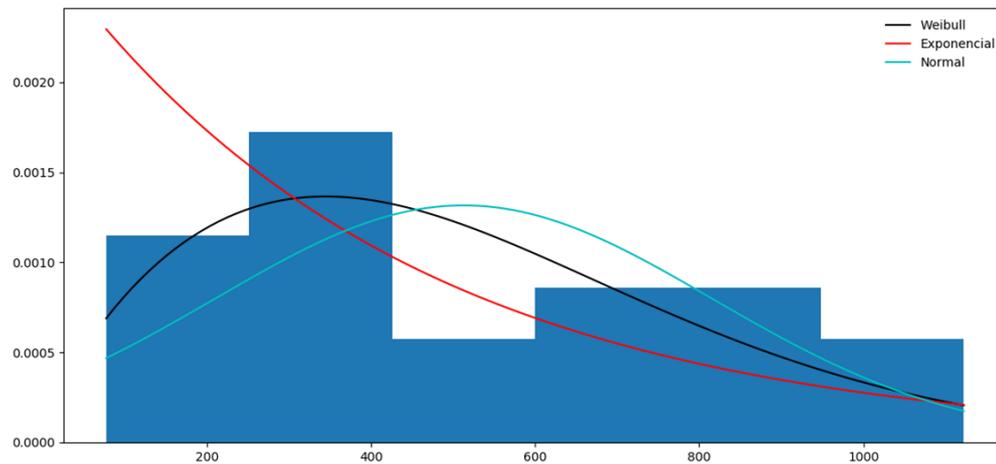


Figura 2.15. Histograma con posibles distribuciones ajustadas

Es posible percibir, con una breve inspección visual, que la distribución de Weibull parece ofrecer la mejor aproximación de los datos del histograma. El analista hará mejores previsiones a partir de esa distribución, comparada con las demás ajustadas (es decir, normal y exponencial). Al analista le resta obtener los mejores valores de parámetros para la distribución de Weibull que le permitan hacer un ajuste óptimo para los datos empíricos, constantes en el histograma.

El objetivo de esta sección será introducir al analista algunas técnicas de estimación de parámetros.

2.3.1 Criterios para estimativas

Suponga que un determinado proceso, con una cantidad muy grande de componentes a ser examinados (por ejemplo, un lote de 100000 tornillos), posee una cierta cantidad de componentes dañados. Históricamente, el fabricante estimó que su sistema es capaz de producir piezas adecuadas el 90% de las veces.

El usuario comprador de 100000 piezas necesita estimar cuál porcentaje de tornillos está fuera de conformidad. Para eso, él debe hacer un muestreo con tamaño n , para $n \ll 100000$. Entonces, surgen las siguientes preguntas:

- ¿Cuál es el tamaño de la muestra que permitirá un resultado estimativo satisfactorio?
- ¿Cuánto las estadísticas de la muestra estarán cercanas o alejadas de las estadísticas de la población?
- ¿Un muestreo es suficiente? ¿O serán necesarios varios muestreos? ¿Los varios muestreos deben ser hechos con o sin reposición?
- ¿Las muestras tendrán la misma distribución de probabilidad de la población?

Una técnica para la obtención del tamaño de la muestra puede ser calculada por la siguiente ecuación [DOD (1966)]:

$$n = \left[\frac{\phi \sigma}{k \bar{x}} \right]^2 \quad (2.29)$$

Donde:

- ϕ - nivel de confianza ($\phi = 1,645$ para 95 % de confianza);
- σ - desviación estándar de la población;
- \bar{x} - media de la población;
- k - precisión deseada para la predicción, dada como un porcentaje de la media de la población.

La desviación estándar de la población, σ , está relacionada con la desviación estándar de la muestra de tamaño n por la siguiente ecuación:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \quad (2.30)$$

Las respuestas a todas las preguntas de la página anterior están fuera del estudio de este libro. Otras técnicas de muestreo pueden ser obtenidas en la norma [ANSI/ASQ Z1.4-2003 (2013)] y en la norma [ANSI/ASQ Z1.9-2003 (2013)]. Para el uso de esas normas, el analista debe tener en mente que una inspección por atributos es aquella que verifica si una muestra está “conforme” o “no conforme”, según una condición preestablecida (un patrón constante en una normativa técnica, por ejemplo). Ya una inspección por variables está basada en medidas numéricas de conformidad. Es decir, los datos no son del tipo “*go/no go*”, pero están comprendidos en un rango de valores preestablecidos (por ejemplo, $1,00 \pm 0,05 \text{ cm}$).

De cualquier modo, el analista no deberá esperar que una estimativa \hat{p} de un parámetro p venga a tener el mismo valor de p .

Por lo tanto, serán establecidos los siguientes criterios para estimativas [Meyer (1983)]:

Definición 2.3.1 — Estimador de θ . Sea X una variable aleatoria con alguna distribución de probabilidad que dependa de un parámetro desconocido. Sean X_1, X_2, \dots, X_n un muestreo de X y sean x_1, x_2, \dots, x_n los correspondientes valores muestrales. Si $g(X_1, X_2, \dots, X_n)$ es una función de muestreo para la estimación de θ , entonces g es un estimador de θ y la estimativa de θ es escrita como $\hat{\theta} = g(x_1, x_2, \dots, x_n)$.

Definición 2.3.2 — Estimador no tendencioso. Sea $\hat{\theta}$ una estimativa del parámetro desconocido θ asociado con la distribución de la variable aleatoria X . En este caso, $\hat{\theta}$ será un estimador no tendencioso de θ si $\mathbb{E}(\hat{\theta}) = \theta \forall \theta$.

Definición 2.3.3 — Varianza mínima. Sea $\hat{\theta}$ una estimativa no tendenciosa de θ . $\hat{\theta}$ es una estimativa no tendenciosa, de varianza mínima de θ , si para todas las estimativas θ^* en que $\mathbb{E}(\theta^*) = \theta$, $V(\hat{\theta}) \leq V(\theta^*) \forall \theta$. O sea, la mejor estimativa no tendenciosa de θ es aquella en que $\hat{\theta}$ tiene la menor varianza.

Definición 2.3.4 — Coherencia. Sea $\hat{\theta}$ una estimativa de θ . $\hat{\theta}$ es una estimativa coherente de θ si:

$$\lim_{n \rightarrow \infty} Prob[|\hat{\theta} - \theta| > \varepsilon] = 0 \quad \forall \varepsilon > 0 \quad (2.31)$$

O, de manera equivalente:

$$\lim_{n \rightarrow \infty} Prob[|\hat{\theta} - \theta| \leq \varepsilon] = 1 \quad \forall \varepsilon > 0 \quad (2.32)$$

O sea, si n aumenta, $\hat{\theta} \rightarrow \theta$.

Definición 2.3.5 — Mejor estimativa. Sea $\hat{\theta}$ la mejor estimativa no tendenciosa lineal de θ . Entonces:

1. $\mathbb{E}(\hat{\theta}) = \theta$;
2. $\hat{\theta} = \sum_{i=1}^n a_i X_i \rightarrow \hat{\theta}$ es una función lineal de la muestra;
3. $V(\hat{\theta}) \leq V(\theta^*)$ donde θ^* es cualquier otra estimativa de θ que satisfaga a los numerales 1 y 2.

■ **Ejemplo 2.1** En este ejemplo, haremos una simulación computacional con los datos del problema de los tornillos y aplicaremos los conceptos definidos en las definiciones 2.3.1 a 2.3.5. Por tratarse de una simulación de números al azar, a cada ejecución del programa, los resultados numéricos serán distintos de los anteriores. Eso es lo que ocurre en una situación real, probabilística. Se debe considerar que la computadora simula números pseudoaleatorios.

El algoritmo tendrá las siguientes operaciones:

- Simular un lote de 100 mil tornillos, en el que 10% del total serán defectuosos (o sea, serán señalados con el número 1);
- Sacar una muestra de 100 tornillos al azar del lote del fabricante;
- Sacar una segunda muestra igual;
- Obtener la media y la desviación estándar de las muestras y comparar con los datos del fabricante, en acuerdo con las ecuaciones de la Sección 2.2.7;
- Decidir por cual será el mejor estimador a ser empleado por el analista.

El estimador suministrado por el fabricante es $p = 0,1$ tornillos defectuosos. Luego, en 100 mil tornillos, el valor esperado de tornillos defectuosos es $np = 10000$.

Pero la información de la media es apenas parte del análisis. Es necesario estimar la desviación estándar del proceso desarrollado por el fabricante. En esto caso, tenemos $\sqrt{np(1-p)} = 94,87$.

```

# -*- coding: utf-8 -*-
import numpy as np
from numpy.random import binomial
from numpy.random import randint
import unicodedata
#%%
def med(n,p):
    return n*p
def sd(n,p):
    return np.sqrt(n*p*(1-p))
#%%
p=0.1
tot=100000
medfab=med(tot ,p)
sdfab=sd(tot ,p)
print('La media teorica de productos defectuosos del fabricante es ' +
      unicodedata.lookup("GREEK SMALL LETTER MU") + '=%.01f' %medfab)
print('La desviacion estandar teorica de productos defectuosos del
      fabricante es '+ unicodedata.lookup("GREEK SMALL LETTER SIGMA") + '=%.01f
      ' %sdfab)
#%%
fab=binomial(1, p, tot)
n=100
mues1=np.zeros(n)
mues2=np.zeros(n)
for i in range (n):
    mues1[i]=fab[randint(tot)]
    mues2[i]=fab[randint(tot)]
p1=sum(mues1)/n
p2=sum(mues2)/n
print('La media teorica de productos defectuosos de la muestra 1 es ' +
      unicodedata.lookup("GREEK SMALL LETTER MU") + '=%.01f' %med(n, p1))
print('La desviacion estandar teorica de productos defectuosos de la muestra
      1 es ' + unicodedata.lookup("GREEK SMALL LETTER SIGMA") + '=%.01f' %sd(n
      ,p1))
print('Los valores estimados para la muestra 1 son: p1=%.2f y V1=%.5f' %(p1
      ,(sd(n, p1)/n)**2))
print('La media teorica de productos defectuosos de la muestra 2 es ' +
      unicodedata.lookup("GREEK SMALL LETTER MU") + '=%.01f' %med(n, p2))
print('La desviacion estandar teorica de productos defectuosos de la muestra
      2 es ' + unicodedata.lookup("GREEK SMALL LETTER SIGMA") + '=%.01f' %sd(n
      ,p2))
print('Los valores estimados para la muestra 2 son: p2=%.2f y V2=%.5f' %(p2
      ,(sd(n, p2)/n)**2))

```

Código 2.17: Simulación de tornillos dañados

El Código 2.17 fue ejecutado y retornó los siguientes resultados:

- La media teórica de productos defectuosos del fabricante es $\mu = 10000,0$
- La desviación estándar teórica de productos defectuosos del fabricante es $\sigma = 94,9$
- La media teórica de productos defectuosos de la muestra 1 es $\mu = 9,0$
- La desviación estándar teórica de productos defectuosos de la muestra 1 es $\sigma = 2,9$
- Los valores estimados para la muestra 1 son: $p_1 = 0,09$ y $V_1 = 0,00082$
- La media teórica de productos defectuosos de la muestra 2 es $\mu = 13,0$
- La desviación estándar teórica de productos defectuosos de la muestra 2 es $\sigma = 3,4$
- Los valores estimados para la muestra 2 son: $p_2 = 0,13$ y $V_2 = 0,00113$

Basado en los datos simulados, se puede observar que muestras sacadas al azar de una gran población tuvieron una probabilidad de tornillos dañados muy cercana de los valores estimados por el fabricante. Pero el valor 0,09, referente a la muestra 1, es más cercano de 0,10 (del fabricante), comparado con el valor 0,13 de la muestra 2. En este caso, se puede percibir que el mejor estimador para tornillos fallidos (9%) también presentó una desviación estándar menor, lo que implica también en una varianza menor (0,00082 contra 0,00113), siendo el mejor estimador inicial para el porcentaje de tornillos fallidos.

El código presentado está considerando un muestreo con reposición.

■

2.3.2 Método de la máxima similitud

Otra técnica más general de estimativa de parámetros, es conocida como método de la máxima similitud.

A fin de establecer una notación, consideraremos la pdf de una función siendo $f(x, \theta)$, sea esa pdf continua (calculada en el punto x) o discreta ($P(X = x)$).

Definiremos la función de máxima similitud (L) siendo:

$$L(X_1, \dots, X_n; \theta) = f(X_1; \theta)f(X_2; \theta)\dots f(X_n; \theta) \quad (2.33)$$

El objetivo de la función de máxima similitud es obtener el valor de $\hat{\theta}$ que maximice L . Es decir, si $L(X_1, \dots, X_n; \theta_1) < L(X_1, \dots, X_n; \theta_2)$, el valor de θ_2 será el preferido.

El valor de θ puede ser representado por un vector, una vez que las estadísticas que definen una distribución pueden tener múltiples parámetros – como, por ejemplo, ocurre con la distribución normal $N(\mu, \sigma^2)$.

Si el objetivo es obtener un valor máximo para la función L , un método de cálculo puede ser empleado para facilitar la obtención del valor máximo. Por lo tanto, haremos la siguiente operación matemática:

$$\text{Sea } \ln L(X_1, \dots, X_n; \theta) \Rightarrow \frac{\partial}{\partial \theta} \ln L(X_1, \dots, X_n; \theta) = 0 \quad (2.34)$$

La Ec. (2.34) es conocida como ecuación de máxima similitud.

■ **Ejemplo 2.2** Sea un sistema que posea un tiempo de duración entre fallas (T), que es bien modelado por una distribución exponencial de parámetro β . Entonces:

$$f(t) = \beta e^{-\beta t} \quad \forall t \geq 0 \quad (2.35)$$

Cuando n componentes del sistema son ensayados hasta sus fallas, varios tiempos son computados (T_1, T_2, \dots, T_n). Eso resulta en:

$$L(T_1, T_2, \dots, T_n; \beta) = \beta^n e^{-\beta \sum_{i=1}^n T_i} \quad (2.36)$$

$$\ln L = n \ln \beta - \beta \sum_{i=1}^n T_i \Rightarrow \frac{\partial}{\partial \beta} \ln L = \frac{n}{\beta} - \sum_{i=1}^n T_i = 0 \quad (2.37)$$

De la Ec. (2.37) se puede obtener el valor de $\hat{\beta} = 1/\bar{T}$, donde \bar{T} es el tiempo medio entre fallas.

Para simular ese sistema, será presentado el código 2.18. El código simulará 1000 tiempos de falla al azar, según la distribución representada por la Ec. (2.35). Entonces, numéricamente, se desea obtener el mismo valor analítico para el tiempo medio entre fallas. Para el caso simulado, será asumido que $\beta = 0,1 h^{-1}$, lo que implicará en un tiempo medio entre fallas de aproximadamente 10 horas. La media empírica obtenida fue de 9,8 horas, para la simulación ejecutada.

```
# -*- coding: utf-8 -*-      1
import numpy as np          2
import matplotlib.pyplot as plt 3
# %%                          4
beta=0.1                    5
t=np.random.exponential(scale=1/beta, size=1000) 6
plt.hist(t, label='Media(t)=%.1f horas' %np.mean(t)) 7
plt.xlabel('Tiempos de fallas') 8
plt.ylabel('Frecuencias de fallas') 9
plt.legend(loc='best', frameon=False) 10
plt.show()                  11
```

Código 2.18: Simulación de fallas de un sistema

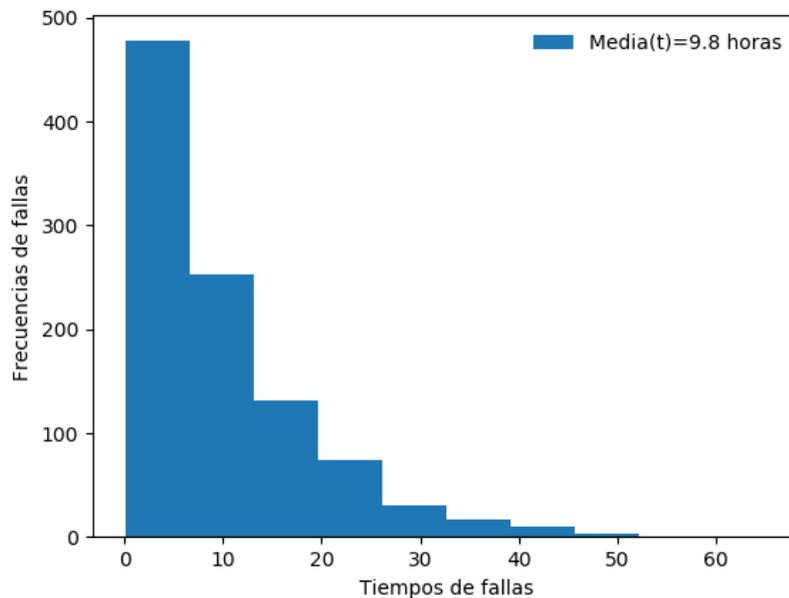


Figura 2.16. Simulación de fallas de un sistema



2.3.3 Método de los mínimos cuadrados

Sea un sistema que puede estar bien modelado por una ecuación lineal, por ejemplo. Sus datos se presentan de una manera lineal, pero con una variabilidad natural alrededor de una recta.

Esa variabilidad (ε) puede ser debida a errores de medición, a pequeñas turbulencias en un sistema térmico o fluido, a rugosidades en una superficie, a un ruido eléctrico, así como muchas otras posibles causas de fluctuación en los datos, respecto a un comportamiento determinista.

Dado el sistema del ejemplo, lo podemos modelar con la siguiente ecuación:

$$Y = \alpha X + \beta + \varepsilon \quad (2.38)$$

La variabilidad ε tiene las siguientes características:

$$\mathbb{E}(\varepsilon) = 0; V(\varepsilon) = \sigma^2 \forall X \quad (2.39)$$

El método de los mínimos cuadrados tiene por objetivo ajustar los parámetros α y β de una recta a los datos empíricos, minimizando al máximo la diferencia entre los datos empíricos y los datos ajustados. Por lo tanto, se debe tornar mínima la expresión:

$$S(\alpha, \beta) = \sum_{i=1}^n [Y_i - (\alpha x_i + \beta)]^2 \quad (2.40)$$

Eso se hace tomando las derivadas parciales de S con respecto a sus parámetros, es decir: $\frac{\partial S}{\partial \alpha} = 0$ y $\frac{\partial S}{\partial \beta} = 0$

El mismo método también puede ser aplicado a sistemas no lineales.

Para demostrar una aplicación, se desarrollará un programa para la generación de datos lineales, con una aplicación Python para la determinación de una recta que mejor se ajuste a los puntos generados al azar.

■ **Ejemplo 2.3** Sea un sistema lineal descrito por una función $y = ax + b$. A ese sistema se suma un ruido blanco de valor ε . Por el método de los mínimos cuadrados, se obtendrá una recta con parámetros tan próximos como sean posibles para la recta determinista.

```

# -*- coding: utf-8 -*-      1
import numpy as np          2
import matplotlib.pyplot as plt  3
from scipy.optimize import curve_fit  4
#%%                          5
a=1                          6
b=2                          7
x=np.linspace(0,10,200)     8
def lin(x,a,b):             9
    return a*x+b           10
y=lin(x,a,b)               11
y1=np.zeros(200)          12
for i in range(200):      13
    y1[i]=y[i]+np.random.uniform(low=-0.5,high=1)  14
popt, pcov = curve_fit(lin, x, y1)  15
y2=lin(x, popt[0], popt[1])  16
plt.plot(x,y, label='Determinista: y=%dx + %d'%(a,b))  17
plt.plot(x,y1, 'ro', markersize=1, label='Empirico')  18
plt.plot(x,y2, label='Minimos cuadrados: y=%2fx + %2f'%(popt[0], popt[1]))  19
plt.xlabel('x')           20
plt.ylabel('y')           21
plt.legend(loc='best', frameon=False)  22
plt.show()                23

```

Código 2.19: Simulación una curva lineal con ruido

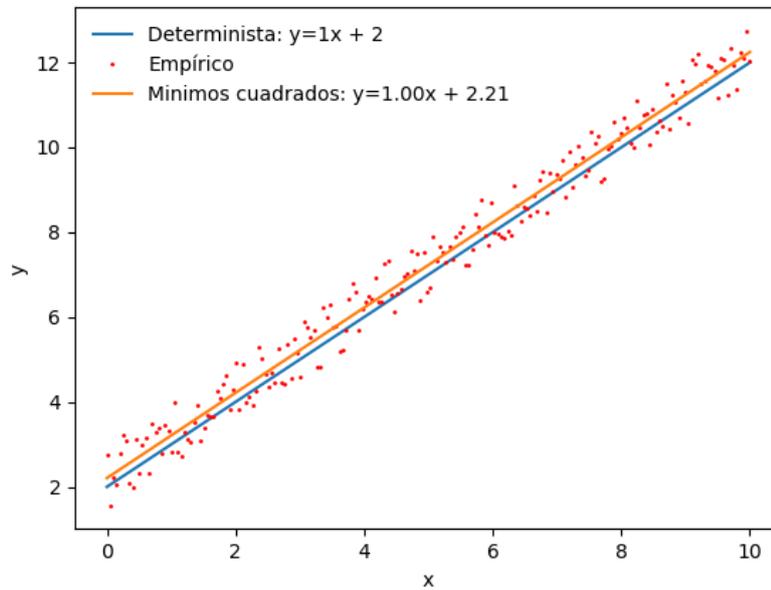


Figura 2.17. Ajuste de una curva lineal por el método de los mínimos cuadrados

Observe que la curva ajustada se quedó un poco arriba de la curva determinista. Eso ocurrió porque el ruido fue configurado para tener una variabilidad uniforme en el intervalo $-0,5 \leq \varepsilon \leq 1$. Eso implicó en valores sumados más grandes y valores sustraídos más pequeños. El lector puede repetir la simulación con un ruido blanco siguiendo una distribución normal $N(0,0.1)$ y percibir que, como la distribución es simétrica, los puntos arriba de la reta determinista serán distribuidos de manera equivalente a los puntos abajo, haciendo coincidir las curvas determinista y ajustada.

■

2.3.4 Test de Kolmogorov-Smirnov

El test de Kolmogorov-Smirnov (K-S) compara una cdf empírica (ECDF) con una cdf de una determinada distribución. Cuanto menor es la distancia entre la cdf empírica y la cdf ajustada, mejor será el ajuste.

■ **Ejemplo 2.4** En este ejemplo, serán simulados 100 puntos, al azar, que están distribuidos según una distribución normal $N(45,16)$. Para estos puntos, será establecida una cdf empírica y comparada con 2 distribuciones normales: $N(45,16)$ y $N(30,100)$. Se espera que el test K-S ofrezca valores más bajos para la comparación con la distribución $N(45,16)$.

```

# -*- coding: utf-8 -*-
from scipy import stats
from scipy.stats import norm
import matplotlib.pyplot as plt
import numpy as np
from statsmodels.distributions.empirical_distribution import ECDF
#%%
ks=np.random.normal(loc=45, scale=4, size=100)
ecdf = ECDF(ks)
x1 = np.linspace(norm.ppf(0.01, loc=45, scale=4), norm.ppf(0.99, loc=45,
scale=4), 100)
x2 = np.linspace(norm.ppf(0.01, loc=30, scale=10), norm.ppf(0.99, loc=30,
scale=10), 100)
plt.plot(x1, norm.cdf(x1, loc=45, scale=4),label='$\mu=45$ y $\sigma=4$')
plt.plot(x2, norm.cdf(x2, loc=30, scale=10),label='$\mu=30$ y $\sigma=10$')
plt.plot(ecdf.x,ecdf.y,'r',label='(ECDF) $\mu=45$ y $\sigma=4$')
plt.title("Distribucion Normal (cdf)")
plt.xlabel("x")
plt.ylabel("Probabilidad")
plt.legend(loc='best',frameon=False)
plt.show()
#%%Comparando dos distribuciones
d1=np.random.normal(45, 4, 100)
d2=np.random.normal(30, 10, 100)
if stats.ks_2samp(ks,d1)[1]>stats.ks_2samp(ks,d2)[1]:
    print('La distribucion con media 45 y desviacion estandar 4 es mejor')
else:
    print('La distribucion con media 30 y desviacion estandar 10 es mejor')

```

Código 2.20: Comparación entre dos distribuciones por el test K-S

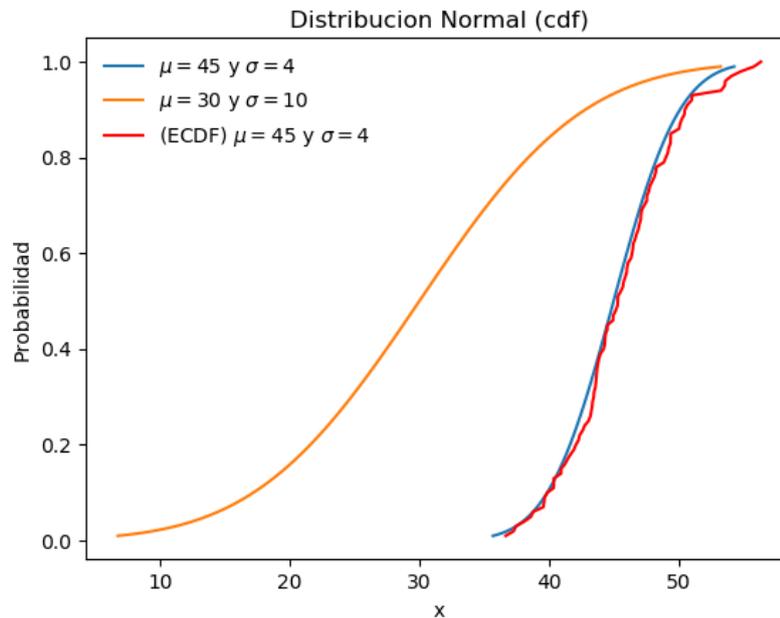


Figura 2.18. ECDF y cdf de 3 distribuciones normales

Observe que la ECDF de los datos simulados se quedó coincidente con la distribución normal $N(45,16)$ y bien distante de la $N(30,100)$. La estadística de K-S para la normal $N(45,16)$ fue $D=0,06$ y para $N(30,100)$ fue $D=0,78$. Esto demuestra que la $N(45,16)$ está mucho más cercana de la ECDF simulada. Además de eso, el valor-p de la comparación con la $N(45,16)$ fue de 99 %, y de la comparación con la $N(30,100)$ fue del orden de 10^{-28} . Esto demuestra que pequeños valores-p (abajo de 5 % o tan pequeños cuanto 1/1000) son indicativos que los datos no pertenecen a la misma distribución de referencia.



2.3.5 Criterio de información de Akaike

El criterio de Akaike (AIC), desarrollado por el matemático japonés Hirotugu Akaike (1927-2009), es un método basado en la Teoría de la Información de Shannon que *grosso modo* estima la cantidad de información perdida si es empleado un determinado modelo de probabilidad. Si dos modelos son empleados, lo que más pierde información debe ser rechazado.

La ecuación de Akaike es la siguiente:

$$AIC = 2k - 2 \ln \hat{L} \quad (2.41)$$

La Ec. (2.41) posee un término k , que se refiere al número de parámetros estimados del modelo, y un término \hat{L} , que es el máximo valor de la función de la máxima similitud, presentada en la Sección 2.3.2.

Cuando el conjunto de datos es pequeño o la distribución tiene muchos parámetros, la Ec. (2.41) cambia para una forma corregida, presentada por la Ec. (2.42), siendo n el número de muestras:

$$AIC_c = AIC + \frac{2k^2 + 2k}{n - k - 1} \quad (2.42)$$

■ **Ejemplo 2.5** En este ejemplo, serán simulados 100 puntos al azar que están distribuidos según una distribución normal $N(10,1)$. Para estos puntos, será establecida una pdf empírica y comparada con 2 otras distribuciones normales: $N(15,9)$ y $N(11,4)$. Se espera que el criterio de Akaike ofrezca valores más bajos para la comparación con la distribución $N(11,4)$.

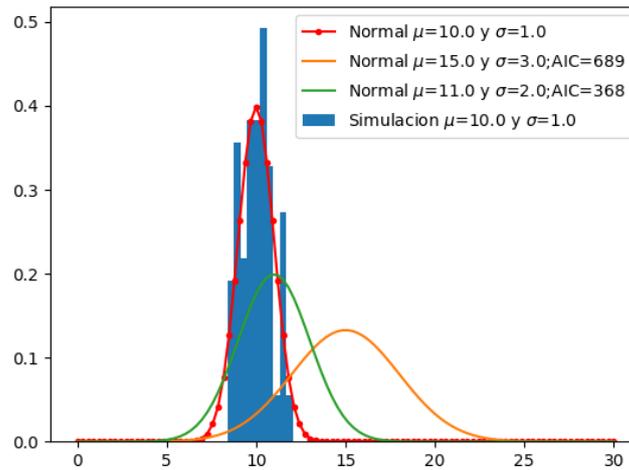


Figura 2.19. Comparación entre dos distribuciones por el criterio de Akaike

Observe que la curva ajustada (en rojo) se quedó sobre el histograma de los puntos simulados. Visualmente, la curva con parámetros más cercanos ($N(11,4)$, en verde) es la candidata que mejor representaría el sistema simulado. Se observa que su valor de AIC fue más pequeño que aquel de la distribución $N(15,9)$.

Para 100 puntos, la Ec. (2.42) sería más adecuada. Pero, al compararse 2 distribuciones con las mismas características (n y k iguales), a los términos de la ecuación corregida serían sumados los mismos valores para ambas ecuaciones. En ese caso, no harían diferencia. El analista debe dedicar especial atención cuando las distribuciones sean diferentes y/o cuando tengan cantidades de muestras diferentes.

```

# -*- coding: utf-8 -*-
import numpy as np
from numpy.random import normal
import matplotlib.pyplot as plt
from scipy.stats import norm
#%%
def likenorm(x, mu, sig):
    t1 = (1 / ((2 * np.pi * (sig ** 2)) ** (len(x) / 2)))
    t2 = 0
    for i in range(len(x)):
        t2 = t2 + (x[i] - mu) ** 2
    t3 = np.exp(-t2 / (2 * (sig ** 2)))
    return t1 * t3
def aic(k, l):
    return 2 * k - 2 * np.log(l)
#%%
k = 2
n = 100
mudat = 10
sigdat = 1
muref1 = 15
sigref1 = 3
muref2 = 11
sigref2 = 2
dat = normal(mudat, sigdat, n)
l1 = likenorm(dat, muref1, sigref1)
l2 = likenorm(dat, muref2, sigref2)
aic1 = aic(k, l1)
aic2 = aic(k, l2)
if aic1 < aic2:
    print("Los parametros que mas se acercan de \u03BC=%.1f y \u03C3=%.1f
          son \u03BC=%.1f y \u03C3=%.1f" %(mudat, sigdat, muref1, sigref1))
else:
    print("Los parametros que mas se acercan de \u03BC=%.1f y \u03C3=%.1f
          son \u03BC=%.1f y \u03C3=%.1f" %(mudat, sigdat, muref2, sigref2))
#%%
x = np.linspace(0, 30, n)
plt.hist(dat, density=True, label="Simulacion $\mu$=%.1f y $\sigma$=%.1f" %(
    mudat, sigdat))
plt.plot(x, norm.pdf(x, loc=mudat, scale=sigdat), "r.-", label="Normal $\mu$=%.1f y
    $\sigma$=%.1f" %(mudat, sigdat))
plt.plot(x, norm.pdf(x, loc=muref1, scale=sigref1), label="Normal $\mu$=%.1f y
    $\sigma$=%.1f; AIC=%d" %(muref1, sigref1, aic1))
plt.plot(x, norm.pdf(x, loc=muref2, scale=sigref2), label="Normal $\mu$=%.1f y
    $\sigma$=%.1f; AIC=%d" %(muref2, sigref2, aic2))
plt.legend(loc="best")
plt.show()

```

Código 2.21: Comparación entre 4 distribuciones por el criterio de Akaike





3. Ingeniería de Confiabilidad

La Ingeniería de Confiabilidad, tema principal de este libro, es una combinación de técnicas proactivas. Involucra rediseño, modificación o mejora de componentes o sus sustituciones por componentes de características superiores [NASA (2000)].

Además, la confiabilidad es una meta que un sector productivo debe perseguir, porque productos de baja confiabilidad presentan daños a intervalos muy cortos e implican insatisfacción por parte del usuario. La reputación de una empresa depende fuertemente del buen desempeño de sus productos.

Se debe tener en mente que la calidad de un producto está relacionada con su conformidad a requisitos al inicio del uso [NIST/SEMATECH (2012)], mientras que la confiabilidad está relacionada con el buen funcionamiento durante un determinado tiempo de uso bajo condiciones establecidas por el fabricante.

Según [Dhillon (2002)], la confiabilidad tuvo su inicio como campo de estudios en la década de 1930, cuando los conceptos de probabilidad fueron aplicados en problemas de generación de energía eléctrica. En la II Guerra Mundial, los conceptos de confiabilidad fueron usados para el desarrollo de los misiles V1 y V2. En la década de 1950, fue adoptada por el United States Department of Defense (DoD) y fue creado un grupo consultivo de confiabilidad de equipos electrónicos (del inglés, *Advisory Group on the Reliability of Electronic Equipment* (AGREE)), que publicó un informe técnico en 1957.

En 1954, hubo un simposio de confiabilidad y control de calidad en los EE.UU., por primera vez, y en 1956 fue publicado el primer libro comercial respecto la confiabilidad en equipos electrónicos. En 1957 la Fuerza Aérea Estadounidense (USAF) lanzó su primera normativa de especificación militar de confiabilidad y en 1962 empezó el primer programa de ingeniería de confiabilidad en el Instituto de Tecnología de la Fuerza Aérea (Dayton, Ohio).

3.1 Cálculos de confiabilidad

En el Capítulo 2, fueron presentados algunos de los modelos de distribución del tiempo de vida.

Sea, por ejemplo, la pdf de una distribución normal, ilustrada por la Fig. 2.1. Considerándose el área bajo la curva como la cdf (es decir, la $F(t)$, ahora una función definida respecto el tiempo),

y asumiéndose que el eje horizontal (de las abscisas) hace referencia al tiempo, el área $F(t)$, desde un tiempo t_1 hasta un tiempo t_2 , hace referencia a la probabilidad que una unidad sacada al azar presente una falla en ese intervalo. O, también, la proporción de una población representada por ese modelo que presentará fallas en el mismo intervalo.

■ **Ejemplo 3.1** Las estimativas históricas de fallas de un determinado sistema apuntan para una buena representación de un tiempo de vida de esto sistema si modelado por una distribución de Weibull, con parámetro de forma $\gamma = 1,5$ y un parámetro de escala de $\alpha = 8000 h$. Si, por año, ese sistema es empleado por 800 horas, ¿cuál será la probabilidad de su falla en 5 años?

```

# -*- coding: utf-8 -*-
import numpy as np
from scipy.stats import weibull_min
import matplotlib.pyplot as plt
#%%
t=5*800
gam=1.5
alf=8000
print("La probabilidad de falla en %d horas es de %.4f" %(t, weibull_min.cdf(
    t, gam, loc=0, scale=alf)))
#%%
from matplotlib.patches import Polygon
from numpy import trapz

tx = np.linspace(0, 15000,1000)
ft=weibull_min.pdf(tx ,gam, scale=alf)

fig , ax = plt.subplots()
ix = np.arange(0, t, 10)
plt.plot(tx , ft)
verts = [(0, 0), *zip(ix , weibull_min.pdf(ix ,gam, scale=alf)), (t, 0)]
poly = Polygon(verts , facecolor='0.9' , edgecolor='0.5')
ax.add_patch(poly)

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.xaxis.set_ticks_position('bottom')

fi=weibull_min.pdf(ix ,gam, scale=alf)
it=trapz(fi ,dx=10)
plt.text(2000, 0.00004, r"$\int_0^t f(t)\mathrm{d}t$=%.2f" %it ,
    horizontalalignment='center' , fontsize=10)

plt.title("Distribucion de Weibull (pdf)")
plt.xlabel("t")
plt.ylabel("Densidad de probabilidad")
plt.show()
#%%
fft=weibull_min.cdf(tx , gam, loc=0, scale=alf)
plt.figure()
plt.title("Distribucion de Weibull (cdf)")
plt.plot(tx , fft)
plt.hlines(weibull_min.cdf(t , gam, loc=0, scale=alf),0,4000,'r')
plt.vlines(4000,0,weibull_min.cdf(t , gam, loc=0, scale=alf),'r')
plt.xlabel("t")
plt.ylabel("Probabilidad")
plt.show()

```

Código 3.1: Cálculo de la probabilidad de falla en una distribución de Weibull

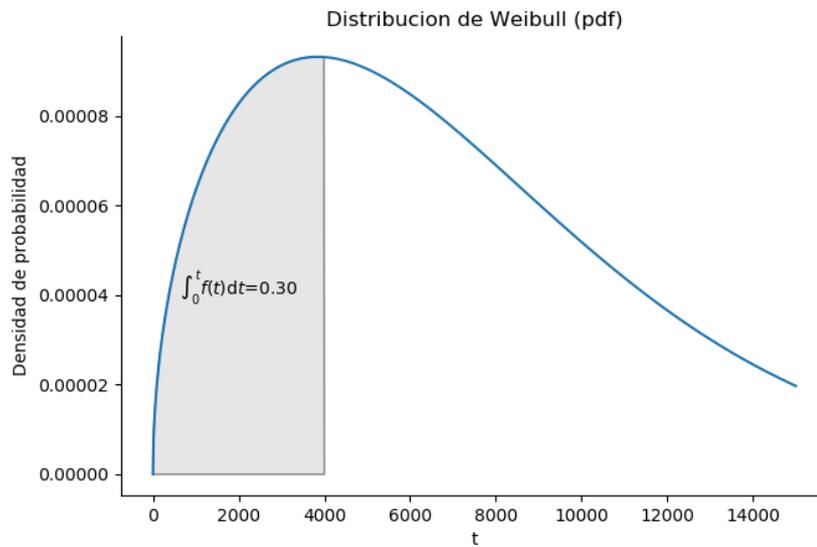


Figura 3.1. Distribución de Weibull (pdf)

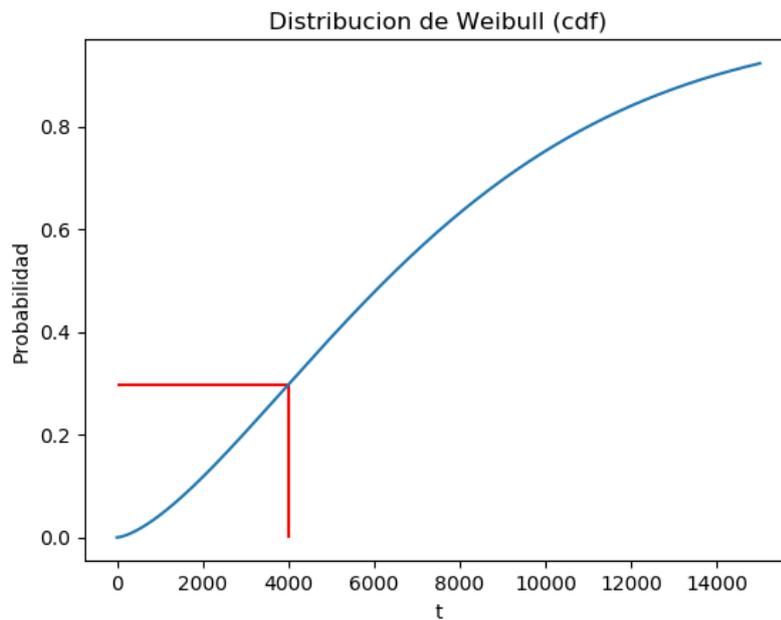


Figura 3.2. Distribución de Weibull (cdf)

El resultado fue:

- La probabilidad de falla en 4000 horas es de 0.2978

Las gráficas 3.1 y 3.2 fueron generadas con la finalidad de ilustrar los cálculos.

■

La función de confiabilidad, presentada en la Ec. (2.9), indica la probabilidad de que una unidad sobreviva más allá de un tiempo t . Algunas veces, es conocida como función de supervivencia ($S(t)$).

La función tasa de fallas, presentada por la Ec. (2.10), es definida para poblaciones no reparables e indica la tasa de fallas de sobrevivientes hasta el tiempo t , durante el próximo instante de tiempo. Es una tasa de fallas instantánea.

La función de riesgo acumulativa, Ec. (2.11), es útil para el cálculo de una tasa media de fallas (AFR) en un intervalo $t_1 \rightarrow t_2$. También determina la tasa media de fallas después del tiempo t . Siendo la integral de $h(t)$ en el intervalo $0 \leq T \leq t$, la Ec. (2.11) puede también ser escrita como:

$$H(t) = -\ln R(t) \quad (3.1)$$

En el intervalo $t_1 \leq T \leq t_2$, resulta:

$$AFR(t_2 - t_1) = \frac{\ln R(t_1) - \ln R(t_2)}{t_2 - t_1} \quad (3.2)$$

Asumiéndose el concepto de tasa de fallas, algunos sistemas y componentes presentan fallas relacionadas con su edad, como fue descrito en la Sección 1.4 y presentado en la Fig. 1.1. Esos sistemas poseen 3 fases, con diferentes tasas de fallas: fallas prematuras (o mortalidad infantil), fallas intrínsecas (periodo de fallas estables) y fallas por desgaste. La Fig. 3.3, comprobada al largo de muchos años de colecta de datos empíricos, presentará este comportamiento físico.

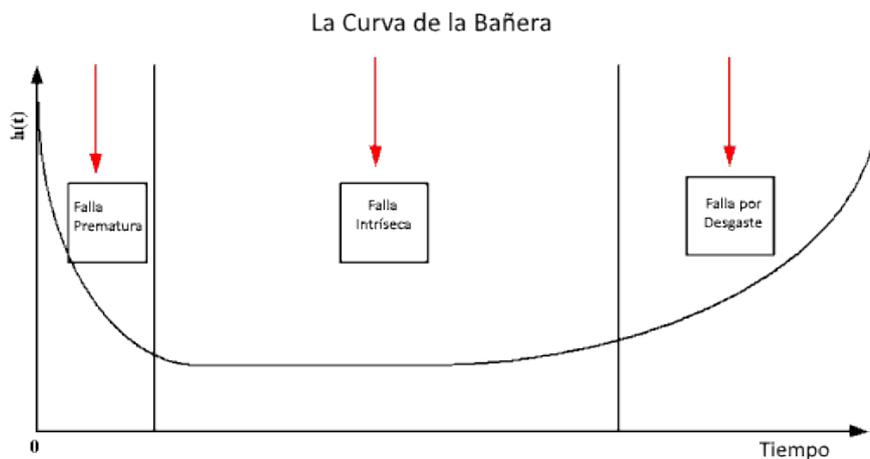


Figura 3.3. Curva de la bañera. Copyright 2012 por National Institute of Standards and Technology

3.1.1 Tasa de reparos o ROCOF

Para sistemas reparables, es posible determinar la tasa de ocurrencia de fallas desde su tiempo $t_0 = 0$ hasta el final de su ciclo de vida (conforme definido en la Sección 1.6 y Fig. 1.2).

Considere que cada falla es contabilizada como una unidad que ocurre a un tiempo t de utilización de un sistema. Si es tomada una función cumulativa de esas fallas, entonces esa función empírica será definida como $N(t)$.

Si varios sistemas semejantes son monitoreados al largo de un tiempo, entonces la curva representativa de $N(t)$ de todos los sistemas puede ser obtenida a partir de una media de todas las curvas. Esa curva, designada por $M(t)$, es el número esperado de fallas cumulativas, en un tiempo t , representativa del sistema en estudio [NIST/SEMATECH (2012)].

La derivada de $M(t)$ respecto al tiempo, $\frac{dM(t)}{dt} = m(t)$, es definida como la tasa de reparos, o la tasa de ocurrencia de fallas a un tiempo t (ROCOF).

Una manera de modelar $M(t)$ es por medio de procesos de Poisson, que serán estudiados en seguida.

Proceso de Poisson homogéneo (HPP)

Este modelo considera que la tasa media de fallas es constante, es decir, no es progresiva ni regresiva con el pasar del tiempo.

Al hacer esa consideración, se pueden obtener las siguientes definiciones y usos de ecuaciones ya presentadas en este libro (Sección 2.2.8).

- La cdf, definida como $F(t) = 1 - e^{-\lambda t}$, indica el tiempo entre dos fallas;
- La pdf es definida como: $P[N(T) = k] = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$;
- El número esperado de fallas a través un tiempo t es $M(t) = \lambda t$;
- La tasa de reparos o ROCOF es $m(t) = \lambda$;
- El tiempo medio entre fallas (MTBF) es el inverso de la tasa de reparos, o sea, $MTBF = \frac{1}{\lambda}$.

Por ese modelo, es posible obtener la probabilidad de k fallas en un tiempo t , por medio de la pdf de una distribución de Poisson, con media λt .

Proceso de Poisson no homogéneo, que sigue una ley de potencia (NHPP(PL))

Cuando la tasa de fallas no es constante, es posible modelar su variación al largo de un tiempo por medio de una expresión polinomial.

Así, tendremos las siguientes expresiones de utilidad para el análisis de un sistema que sigue ese modelo.

- El número esperado de fallas a través un tiempo t es $M(t) = at^b \forall a, b > 0$;
- La tasa de reparos o ROCOF es $m(t) = abt^{b-1} = \alpha t^{-\beta} \forall \alpha > 0, \beta < 1$;
- La pdf es definida como: $P[N(T) = k] = \frac{(M(t))^k}{k!} e^{-M(t)} = \frac{a^k T^{bk} e^{-aT^b}}{k!}$;
- La cdf, definida como $F(t) = 1 - e^{-\int_0^t M(T+\tau) d\tau} = 1 - e^{-a[(T+t)^b - T^b]}$, para una falla ocurriendo al tiempo T .

Según [NIST/SEMATECH (2012)], ese modelo también es conocido por el Modelo de Duane y por Modelo del *United States Army Materials System Analysis Activity* (AMSAA).

Para el primer tiempo de falla, ese proceso presenta una distribución de Weibull con parámetro de forma b y vida característica a .

■ **Ejemplo 3.2** Para simular un proceso NHPP(PL), se sigue el siguiente algoritmo:

1. Se establecen las constantes a y b ;
2. Se genera un vector de n elementos, con una distribución uniforme $U(0,1)$;
3. Se calcula $Y_1 = \left(-\frac{1}{a} \ln U_1\right)^{1/b}$;
4. Se calcula $Y_i = \left(Y_{i-1}^b - \frac{1}{a} \ln U_i\right)^{1/b} \forall i = 2, 3, \dots, n$;
5. Los números generados Y_n siguen una NHPP(PL)

El programa 3.2 genera 13 datos al azar, siguiendo el algoritmo presentado, y traza una curva de ajuste al número esperado de fallas a través del tiempo ($M(t)$).

```

# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt

a=0.2
b=0.4
n=13
t=np.linspace(1,n,num=n)
u=np.random.rand(n)
y=np.zeros(n)
y[0]=(-1/a*np.log(u[0]))
for i in range(1,n):
    y[i]=(y[i-1]**b-1/a*np.log(u[i]))**(1/b)
Mt=a*(y**b)

plt.plot(y,t,label="Simulación")
plt.plot(y,Mt,label="M(t)")
plt.title("Simulación de datos NHPP en Ley de Potencia")
plt.xlabel("Tiempos de falla")
plt.ylabel("Número de fallas acumulativas")
plt.legend(loc='best', frameon=True)
plt.show()

```

Código 3.2: Simulación de datos NHPP en Ley de Potencia

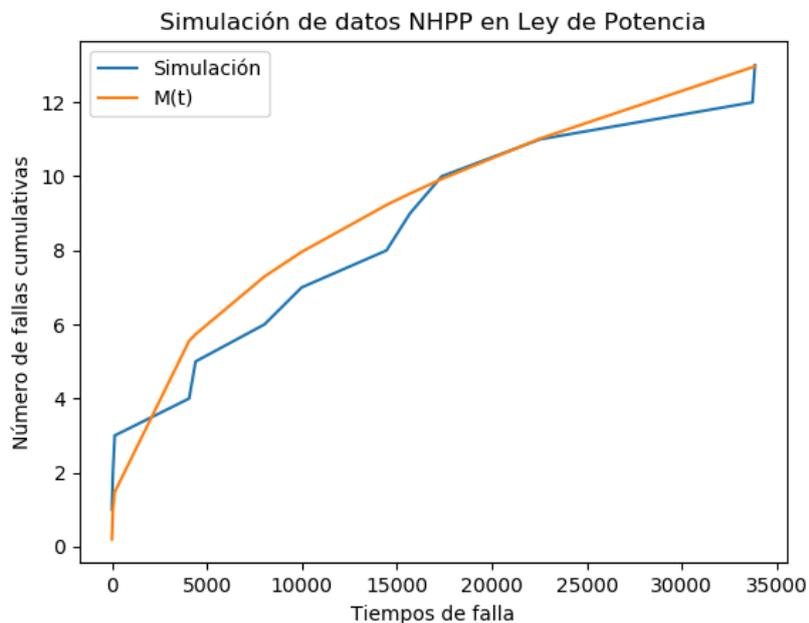


Figura 3.4. Simulación de datos NHPP en Ley de Potencia, con ajuste de una M(t)

■

Proceso de Poisson no homogéneo, que sigue una ley exponencial (NHPP(Exp))

Una NHPP que sigue una ley exponencial posee una ROCOF con la siguiente expresión:

$$m(t) = e^{\alpha + \beta t} \quad (3.3)$$

Así como en la NHPP(PL), cuando $\beta = 0 \forall \alpha = ab$ y $\beta = 1 - b$, el modelo se queda reducido al modelo HPP.

3.1.2 Técnicas de análisis exploratorio de datos

Las técnicas de análisis exploratorio de datos (del inglés, *Exploratory Data Analysis (EDA)*), son métodos de análisis en que se lanzan medios gráficos y cuantitativos para la extracción de informaciones útiles de un conjunto de datos numéricos. En la EDA, es dada énfasis a los análisis gráficos.

Según [NIST/SEMATECH (2012)], las siguientes preguntas son comúnmente hechas por un experto, cuando se depara con datos a analizar:

1. ¿Qué es un valor típico?
2. ¿Cuál es la incertidumbre para un valor típico?
3. ¿Cuál sería una buena distribución ajustada para un conjunto de datos?
4. ¿Cuánto es un percentil?
5. ¿Una modificación de ingeniería tiene un efecto?
6. ¿Un factor tiene un efecto?
7. ¿Cuáles son los factores más importantes?
8. ¿Las mediciones ejecutadas por laboratorios diferentes son equivalentes?
9. ¿Cuál es la mejor función para relacionar una variable de respuesta con un conjunto de variables de factores?
10. ¿Cuáles son los mejores ajustes para los factores?
11. ¿Es posible separar la señal del ruido en un conjunto de datos dependientes del tiempo?
12. ¿Hay alguna estructura en los datos multivariados?
13. ¿Los datos tienen valores atípicos?

En el *handbook* [NIST/SEMATECH (2012)], son presentadas 33 técnicas gráficas de interés de la EDA, que suministran al analista visualizaciones de los datos, para que se puedan descubrir las estructuras subyacentes a los números. En este libro, serán presentadas 4 técnicas gráficas poderosas para el análisis de datos. Otras técnicas (histograma, gráfica de dispersión) descritas en [NIST/SEMATECH (2012)] ya fueron empleadas en este libro y no serán discutidas. Al lector, se deja la sugerencia de profundizar sus conocimientos en técnicas que sean útiles para sus aplicaciones específicas.

La gráfica de autocorrelación

Datos autocorrelacionados son aquellos que poseen alguna correlación con otros datos del mismo conjunto, muchas veces presentando tal correlación de manera recurrente. Es decir, cuando poseen alguna periodicidad, los datos presentan valores de autocorrelación significantes, si son comparados con sus correlatos.

Una técnica para verificar la autocorrelación en datos es útil para evaluar:

- Si los datos son aleatorios;
- Si una observación está relacionada con otra adyacente;
- Si hay ruidos en una serie temporal;
- Si la serie temporal es periódica;
- Si la serie temporal presenta autorregresión;
- Si un modelo es apropiado para una serie temporal;
- Si un modelo del tipo $Y = \text{constante} + \text{error}$ es válido y suficiente;
- Si la fórmula del error estándar $S_Y = \frac{s}{\sqrt{N}}$, citada en la Sección 2.2.3, es válida.

Si adoptamos Y_1, Y_2, \dots, Y_N como las observaciones a los tiempos t_1, t_2, \dots, t_N , la función de autocorrelación es dada por la Ec. (3.4).

$$r_k = \frac{\sum_{i=1}^{N-k} (Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^{N-k} (Y_i - \bar{Y})^2} \quad (3.4)$$

Por ejemplo, los datos de una función sinusoidal presentan correlación positiva máxima con otros valores numéricos distantes a múltiplos de una longitud de onda y correlación negativa máxima con valores distantes a múltiplos de media longitud de onda.

Sin embargo, datos no correlacionados no necesariamente presentan comportamiento aleatorio.

■ **Ejemplo 3.3** En este ejemplo, serán generados 1000 números. El primer conjunto de datos contendrá una secuencia de números según una función sinusoidal. El segundo conjunto, será formado por números al azar distribuidos en acuerdo con una distribución uniforme $U(-1,1)$. Se espera obtener correlación fuerte entre los números del primer conjunto, y ninguna correlación entre los números de la distribución uniforme.

```
# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf
# %%
n=1000
x1=np.linspace(0,6*np.pi,n)
y1=np.sin(x1)
x2=np.linspace(0,n,n)
y2=np.random.uniform(-1,1,size=n)
# %%
plt.subplot(2,1,1)
plt.plot(x1,y1)
plt.title("Funcion sinusoidal")
plt.gca().axes.get_xaxis().set_visible(False)
plt.subplot(2,1,2)
plt.plot(x2,y2)
plt.title("Puntos al azar, U(-1,1)")
plt.show()

plot_acf(y1, title='Autocorrelacion (funcion sinusoidal)')
plot_acf(y2, title='Autocorrelacion (U(-1,1))')
```

Código 3.3: Simulación de datos autocorrelacionados y al azar

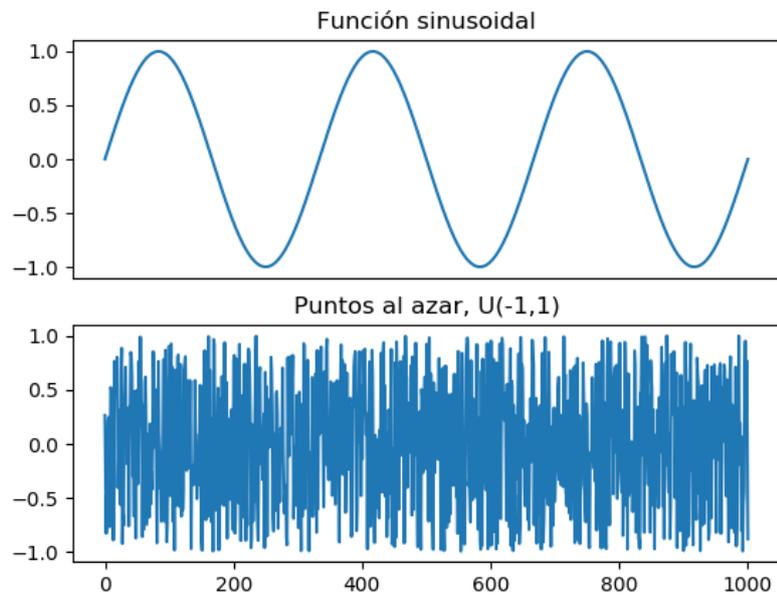


Figura 3.5. Simulación de datos autocorrelacionados y al azar

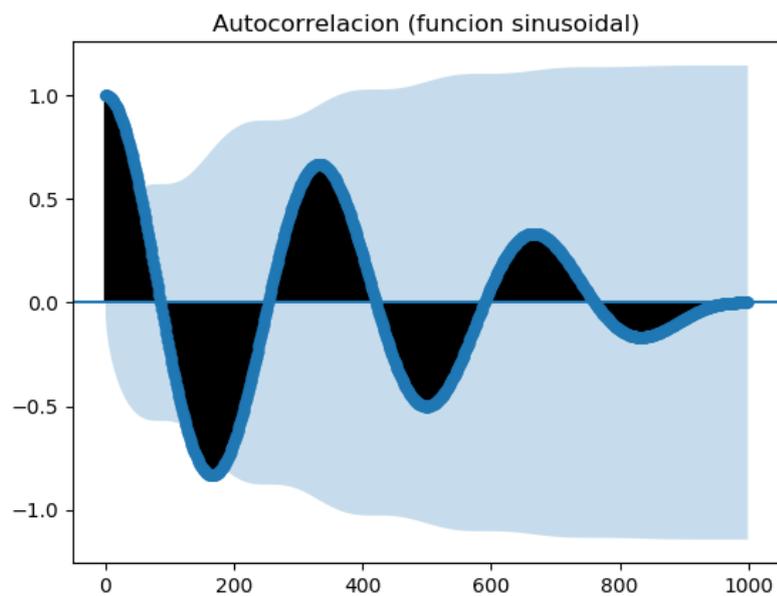


Figura 3.6. Autocorrelación de datos de una función sinusoidal

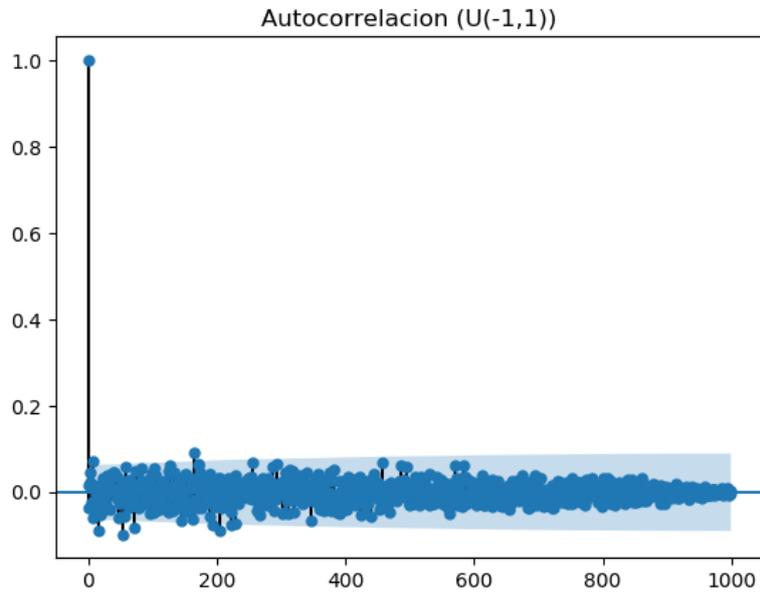


Figura 3.7. Autocorrelación de datos generados al azar



La gráfica de cuantil-cuantil

Esa técnica gráfica permite saber si 2 conjuntos de datos pertenecen a la misma distribución.

La técnica consiste en comparar gráficamente los cuantiles de un conjunto de datos con otro. Entonces, si los puntos de comparación estuvieren sobre una línea de 45 grados en la gráfica, los dos conjuntos de puntos vienen de la misma distribución, con los mismos parámetros.

Es una técnica que debe ser empleada con precaución, porque los números generados al azar pueden pertenecer a distribuciones distintas, pero los cuantiles pueden ser parecidos. Se recomienda que el análisis por esa técnica sea complementado por otros análisis.

■ **Ejemplo 3.4** En este ejemplo, serán generados 100 números al azar. El primer conjunto de datos contendrá una secuencia de números segundo una distribución normal. El segundo conjunto será formado por números al azar según una distribución de Weibull. Ambos los conjuntos serán comparados con una distribución normal (en este caso). Se espera que la comparación entre distribuciones muy diferentes se quede no lineal respecto a una recta.

```

# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
# %%
x1=np.random.normal(loc = 20, scale = 5, size=100)
x2=np.random.weibull(0.5, size=100)

stats.probplot(x1, dist="norm", plot=plt)
plt.title("Grafica cuantil-cuantil (Normal versus Normal)")
plt.show()

plt.figure()
stats.probplot(x2, dist="norm", plot=plt)
plt.title("Grafica cuantil-cuantil (Normal versus Weibull)")
plt.show()

```

Código 3.4: Simulación de datos para la gráfica de cuantil-cuantil

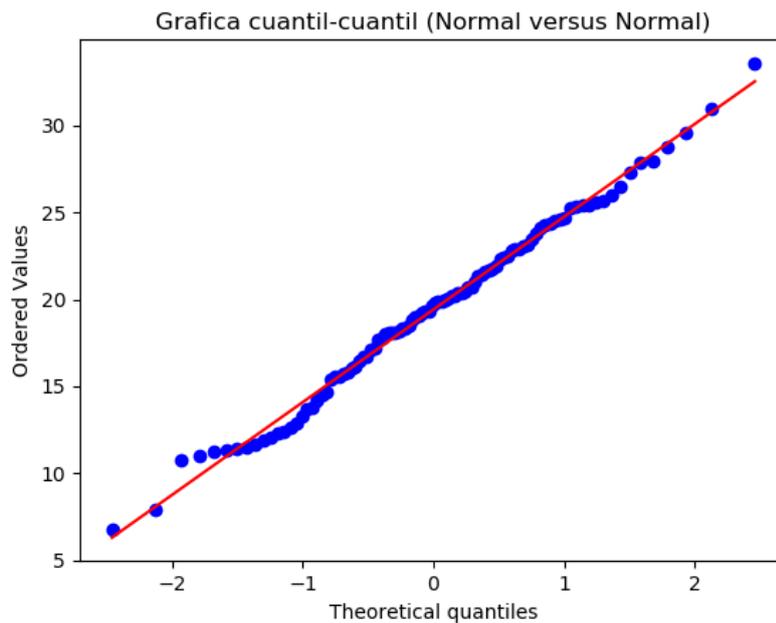


Figura 3.8. Comparación entre 2 distribuciones normales

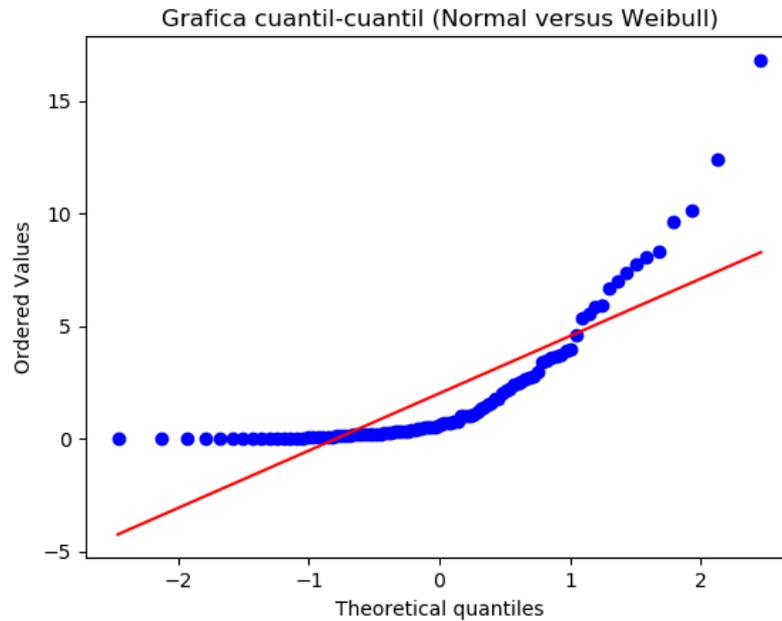


Figura 3.9. Comparación entre una distribución de Weibull e la normal



La gráfica de normalidad de Box-Cox

La técnica de Box-Cox (término en homenaje a sus autores) busca aplicar una transformación matemática a los datos, para que se queden según una distribución normal y, entonces, puedan ser analizados de acuerdo con ese modelo probabilístico.

Se aplican las siguientes transformaciones a los datos no normalizados positivos x_1, \dots, x_n :

$$x(\lambda) = \frac{x^\lambda - 1}{\lambda} \quad \forall \lambda \neq 0 \quad (3.5)$$

$$x(\lambda) = \ln x \quad \forall \lambda = 0 \quad (3.6)$$

El valor adecuado de λ es aquel que maximiza la función de máxima similitud, dada por la Ec. (3.7), donde la media aritmética de la transformada es dada por la Ec. (3.8).

$$f(x, \lambda) = -\frac{n}{2} \ln \left[\sum_{i=1}^n \frac{(x_i(\lambda) - \bar{x}(\lambda))^2}{n} \right] + (\lambda - 1) \sum_{i=1}^n \ln x_i \quad (3.7)$$

$$\bar{x}(\lambda) = \frac{1}{n} \sum_{i=1}^n x_i(\lambda) \quad (3.8)$$

■ **Ejemplo 3.5** En este ejemplo, serán generados 100 números al azar, según una distribución de Weibull.

Será verificado si la muestra generada con la distribución de Weibull se asemeja a una distribución normal, por el método `normaltest()` de la biblioteca Scipy Stats. Este test de normalidad

está basado en los test de D'Agostino y de Pearson. Si el valor-p es menor que el adoptado como referencia ($p = 0,05$) para el test de hipótesis, la distribución no será considerada normal (la hipótesis nula H_0 asume que la distribución es normal).

Una vez comprobada la no normalidad (Fig. 3.10 superior), será aplicada una transformación de Box-Cox, donde podrá ser obtenida la media de los datos transformados. Otras estadísticas podrán ser fácilmente obtenidas a partir de los datos transformados (Fig. 3.10 inferior).

```

# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import normaltest
# %%
a=0.5
x=np.random.weibull(a, size=100)

#Test de hipotesis
stat, p = normaltest(x)
print('Estadisticas=%3f, p=%3f' % (stat, p))
# Interpretacion
alpha = 0.05
if p > alpha:
    print('La muestra es muy semejante a la normal')
else:
    print('La muestra no es semejante a la normal')

fig = plt.figure()
ax1 = fig.add_subplot(211)
prob = stats.probplot(x, dist=stats.norm, plot=ax1)
ax1.set_xlabel('')
ax1.set_title('Grafica cuantil-cuantil (Normal versus Weibull)')

ax2 = fig.add_subplot(212)
xt, lamb = stats.boxcox(x)
prob = stats.probplot(xt, dist=stats.norm, plot=ax2)
ax2.set_title('Grafica cuantil-cuantil (Normal versus Normal Box-Cox)')
# %%
print("La media de la relacion transformada es de %.2f" %np.mean(xt))

```

Código 3.5: Simulación de datos para la gráfica de cuantil-cuantil

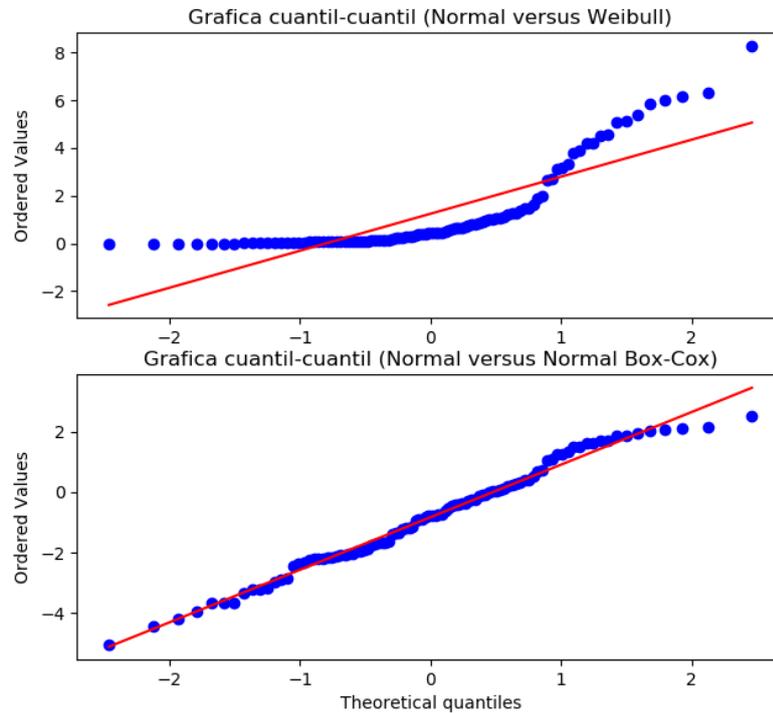


Figura 3.10. Gráfica de cuantil-cuantil para datos transformados



Gráficas de control de Shewhart

En cualquier proceso de producción, siempre existirá una variabilidad intrínseca al proceso, que depende de la contribución de varias causas pequeñas, inevitables, aleatorias. Cuando un proceso opera solamente con variabilidades aleatorias, se dice que se encuentra bajo el control estadístico.

Cuando ocurren variabilidades que son muy grandes comparadas con el nivel natural del sistema, introduciendo tendencias, desplazamientos de la media, o comportamientos de orden no aleatoria, es dicho que el proceso opera en la presencia de causas atribuibles y está fuera de control [Montgomery (2004)].

El fundamento del control estadístico del proceso no tiene como objetivo, necesariamente, eliminar la variabilidad del proceso, pero sí reducir al máximo sus valores.

Una gráfica de control de Shewhart (nombrada en homenaje a Walter Shewhart) es compuesta por una línea central y límites de control, superior e inferior. La posición de esas referencias es generalmente en la media del proceso y a múltiplos de la desviación estándar.

Las gráficas de control son herramientas poderosas para el control de variables o de atributos. [Montgomery (2004)] presenta en su obra una grande cantidad de técnicas para el tratamiento de esos parámetros, con nivel de profundidad que huye al objetivo de este libro. Si el analista siente la necesidad de profundizar sus conocimientos en esas técnicas, [Montgomery (2004)] es una fuente altamente recomendable para el control estadístico de procesos.

Para la interpretación de los datos de control de \bar{x} , es importante proceder a una verificación de conformidad en los gráficos de amplitud de los datos (R). La amplitud de una muestra de tamaño n (x_1, x_2, \dots, x_n) es la diferencia entre su mayor y su menor valor, o sea, $R = x_{max} - x_{min}$.

Si son tomadas varias muestras, el muestreo tendrá una distribución de valores de R que servirá para componer una gráfica de control de R . Cuando el patrón no aleatorio aparece en ambas gráficas (\bar{x} y R), eliminándose las causas atribuibles de la gráfica de R , automáticamente podrán ser eliminadas las causas de las gráficas de \bar{x} .

El Código 3.6 presentará datos simulados de una distribución normal, con sus respectivas referencias: límites superior e inferior de control (LSC y LIC, a 3σ de distancia de la media μ), límites superior e inferior de alerta (LSA y LIA, a 2σ de distancia de la media μ), línea central (media μ). Los límites están destacados en rojo e en amarillo en la Fig. 3.11.

```

# -*- coding: utf-8 -*-      1
import numpy as np          2
import matplotlib.pyplot as plt 3
# %%                          4
n=25                          5
med=10                         6
de=1                            7
nor=np.random.normal(loc=med, scale=de, size=n) 8
x=np.linspace(1, n, num=n)    9
lm=med*np.ones(n)            10
med=np.mean(nor)             11
de=np.std(nor)                12
lsc=med+3*de*np.ones(n)     13
lic=med-3*de*np.ones(n)     14
lsa=med+2*de*np.ones(n)     15
lia=med-2*de*np.ones(n)     16
                                17
plt.plot(x, nor, 'bo')       18
plt.plot(x, nor, 'b')        19
plt.plot(x, lm, 'k')         20
plt.plot(x, lsc, 'r')        21
plt.plot(x, lic, 'r')        22
plt.plot(x, lsa, 'y')        23
plt.plot(x, lia, 'y')        24
plt.show()                   25

```

Código 3.6: Simulación de datos para la gráfica de Shewhart

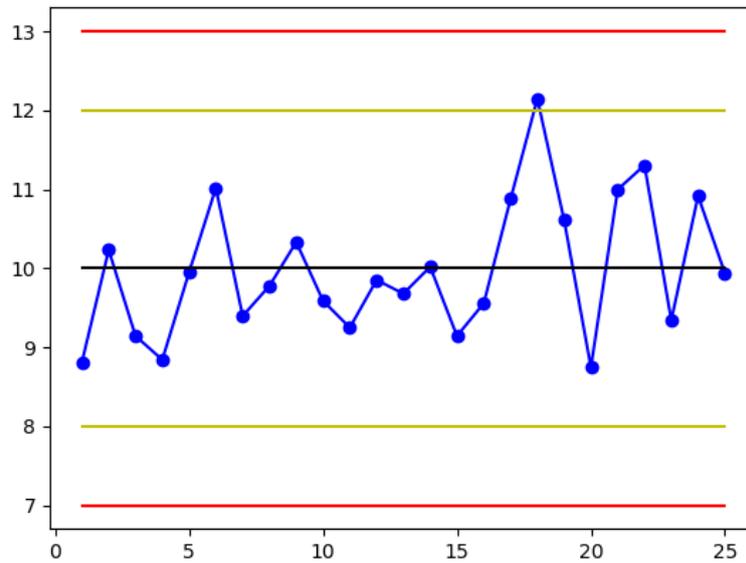


Figura 3.11. Gráfica de Shewhart con $\mu = 10$ y $\sigma = 1$

Una vez que los puntos son normalmente distribuidos alrededor de un valor medio, es posible inferir que 99,7% de los puntos estarán entre los límites superior e inferior de control ($\mu \pm 3\sigma$), 95,4% estarán entre los límites superior e inferior de alerta ($\mu \pm 2\sigma$) y 68,3% estarán entre los límites de una desviación estándar ($\mu \pm 1\sigma$). Puntos más allá de los límites de control son buenos indicadores de fuga de control.

Con el objetivo de analizar posibles indicios de fuga de control del proceso, serán presentados a continuación algunos patrones con sus posibles factores subyacentes, que indican la posible presencia de causas atribuibles.

Los patrones cíclicos, presentados en la Fig. 3.12, pueden representar cambios ambientales sistemáticos, como variaciones de temperatura, fatiga del operador, alternancia de operadores o de máquinas o fluctuaciones en parámetros (tensión eléctrica, presión).

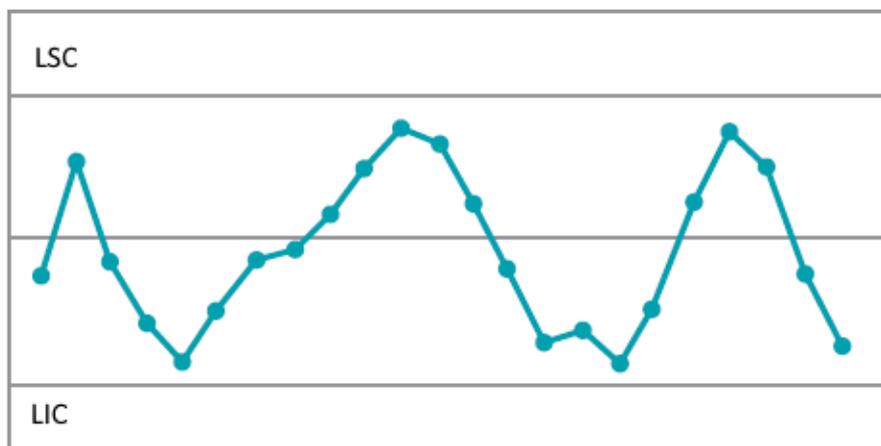


Figura 3.12. Ciclos en una gráfica de control. Copyright 2004 por Livros Técnicos e Científicos

El patrón de mistura, presentado en la Fig. 3.13, se caracteriza por puntos cercanos de los límites. Puede representar la superposición de distribuciones, debida a frecuentes ajustes por parte de los operadores, en respuesta a variables na salida, y no en las causas atribuibles. También pueden deberse a mistura de procesos distintos como, por ejemplo, piezas fabricadas en dos máquinas diferentes y después misturadas, antes del control de calidad.

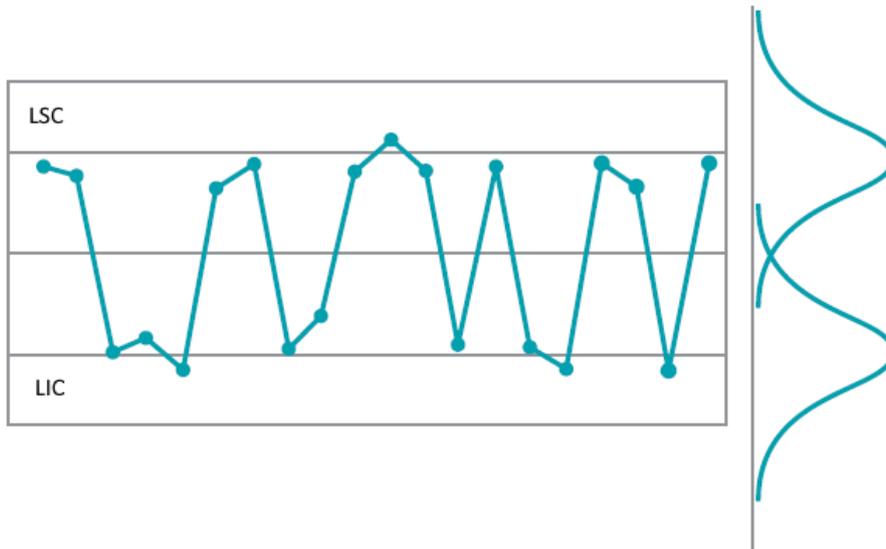


Figura 3.13. Patrón de mistura en una gráfica de control. Copyright 2004 por Livros Técnicos e Científicos

El desplazamiento del nivel del proceso, presentado en la Fig. 3.14, puede ser debido a cambios en procesos productivos, como sustitución de trabajadores, métodos, materiales o máquinas, cambios en procesos de inspección o valores de referencia, cambio en la capacidad laboral de los trabajadores.

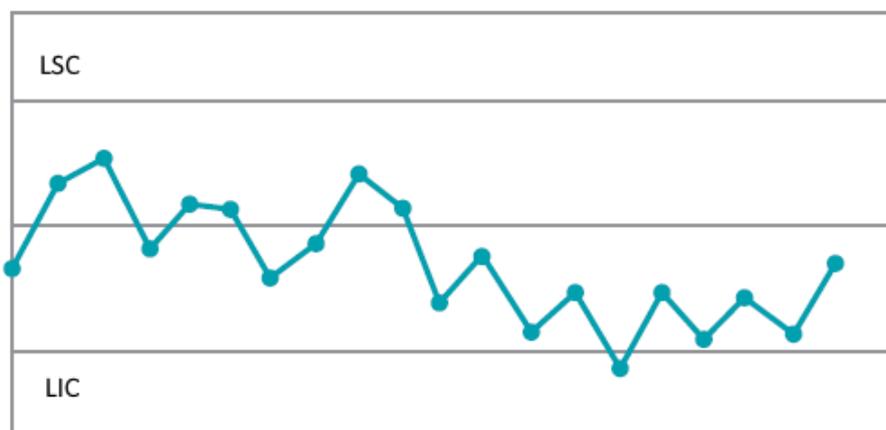


Figura 3.14. Desplazamiento del nivel del proceso. Copyright 2004 por Livros Técnicos e Científicos

La tendencia, presentada en la Fig. 3.15, puede relacionarse con desgaste o deterioración gradual de un equipo.

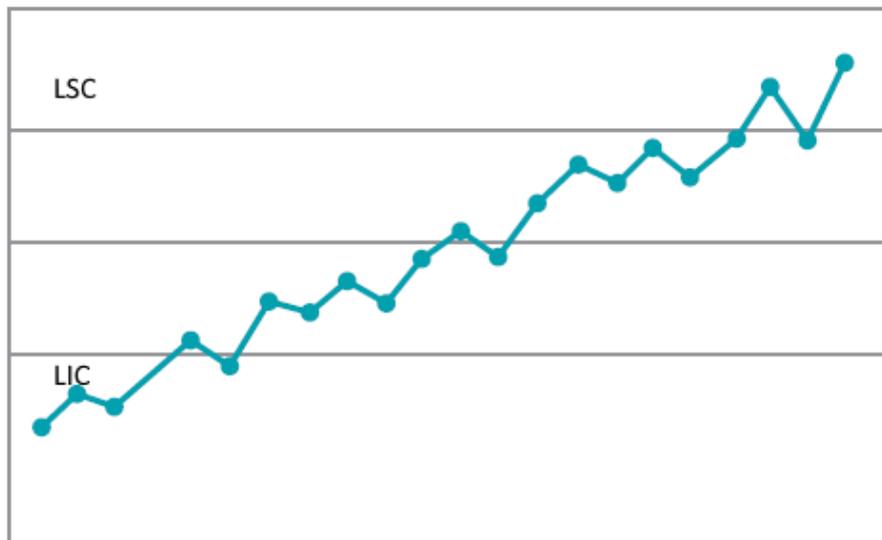


Figura 3.15. Tendencia en una gráfica de control. Copyright 2004 por Livros Técnicos e Científicos

La estratificación, presentada en la Fig. 3.16, es una concentración de los datos alrededor y muy cercanos de la línea central. Puede deberse al cálculo incorrecto de los límites de control o de muestreo de diferentes subgrupos en distribuciones distintas. Por ejemplo, 5 muestras desde 5 procesos diferentes.

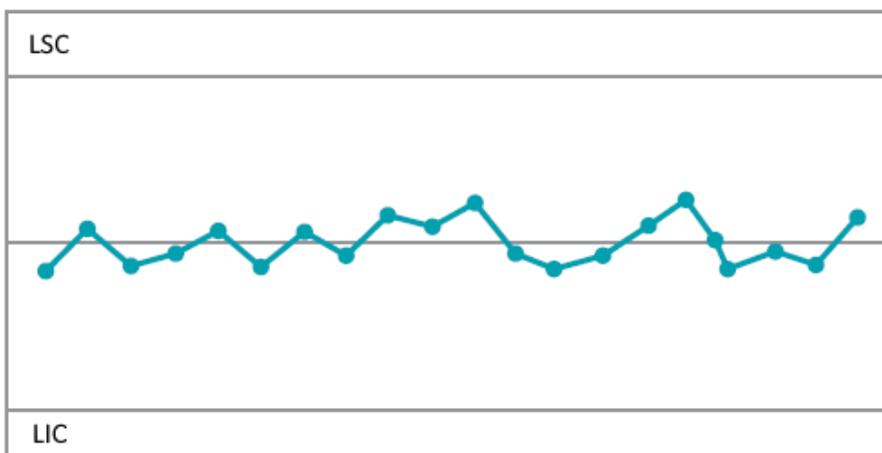


Figura 3.16. Estratificación en una gráfica de control. Copyright 2004 por Livros Técnicos e Científicos

Además de eso, [Montgomery (2004)] presenta 10 reglas sensibilizadoras para las gráficas de control, a saber:

1. Uno o más puntos fuera de los límites de control;
2. Dos o más puntos consecutivos fuera de los límites de alerta 2σ ;
3. Cuatro o cinco puntos consecutivos más allá de los límites 1σ ;
4. Una secuencia de ocho puntos consecutivos de un mismo lado de la línea central;
5. Seis puntos en una secuencia siempre creciente o decreciente;
6. Quince puntos en secuencia en la zona C (arriba o abajo de la línea central);
7. Catorce puntos en secuencia alternativamente para arriba y para abajo;

8. Ocho puntos en secuencia de ambos los lados de la línea central, con ninguno en la zona C;
9. Un patrón no usual o no aleatorio en los datos;
10. Un o más puntos cerca de un límite de alerta o de control.

Para fines de análisis por las reglas sensibilizadoras, se define la zona C como la región entre los límites $\pm 1\sigma$; la zona B como la región entre los límites $\pm 2\sigma$, excluyéndose la zona C; y la zona A como la región entre los límites $\pm 3\sigma$, excluyéndose las zonas B y C.

Una vez automatizado el control de datos por las gráficas de Shewhart, la presencia de patrones no usuales se torna fácil de detectar, permitiendo una acción inmediata por el analista.

3.1.3 Confiabilidad de sistemas

Considere un sistema compuesto por varios subsistemas. Esos subsistemas, cada uno con su confiabilidad inherente, integran una red que origina la confiabilidad del sistema.

Empezando por el comportamiento de los componentes que integran el subsistema, cada uno de los componentes, asumidos sustituibles, posee k diferentes modos de falla, que pueden presentarse como distintos mecanismos de falla [NIST/SEMATECH (2012)].

Un modelo de competencia de riesgo es capaz de evaluar la confiabilidad de cada componente a partir del modelo de confiabilidad para cada modo de falla.

Para ese modelo, serán adoptadas tres hipótesis necesarias:

1. Las fallas son mutuamente independientes;
2. El componente falla en el momento donde el primer mecanismo de falla se presenta en el “estado de falla”; y
3. Cada modo de falla posee una distribución de probabilidad de falla conocida, $F_i(t)$.

Asumiéndose los índices i y c para representar el i -ésimo modo de falla y el componente, respectivamente, las ecuaciones (3.9), (3.10) y (3.11) permiten el cálculo de los modelos de competencia de riesgo, al largo de un tiempo.

$$R_c(t) = \prod_{i=1}^k R_i(t) \quad (3.9)$$

$$F_c(t) = 1 - \prod_{i=1}^k [1 - F_i(t)] \quad (3.10)$$

$$h_c(t) = \sum_{i=1}^k h_i(t) \quad (3.11)$$

■ **Ejemplo 3.6** Sea un componente que presenta 3 modos de falla. Cada modo de falla sigue una ley distinta para su ocurrencia en función del tiempo, a saber:

- Modo de falla I - $N(500, 8100)$;
- Modo de falla II - Weibull, con $\gamma = 0,7$, $\mu = 0$ y $\alpha = 1500$; y
- Modo de falla III - binomial, con media $\mu = 600$ fallas, al largo de 1000 horas de funcionamiento.

Para hacerse un estudio de confiabilidad de ese componente, funcionando por 1000 horas, se desea saber su cdf. El Código 3.7 calculará la cdf para cada modo de falla y después aplicará la Ec. (3.10).

```

# -*- coding: utf-8 -*-
import scipy.stats as st
import matplotlib.pyplot as plt
import numpy as np
#%%
n=1000
x = np.linspace(0, n, n)
mu=500
sig=90
mubin=600
c=0.7
y4=np.zeros(n)

def comb(f1, f2, f3):
    return 1-(1-f1)*(1-f2)*(1-f3)

y1 = st.norm.cdf(x, loc=mu, scale=sig)
y2 = st.weibull_min.cdf(x, c, loc=0, scale=1500)
y3 = st.binom.cdf(x, n, mubin/n)

for i in range(n):
    y4[i]=comb(y1[i], y2[i], y3[i])

plt.subplot(2, 2, 1)
plt.title("Modo de falla I (Normal)")
plt.plot(x, y1)
plt.ylabel("Probabilidad")
plt.gca().axes.get_xaxis().set_visible(False)

plt.subplot(2, 2, 2)
plt.title("Modo de falla II (Weibull)")
plt.plot(x, y2)
plt.gca().axes.get_xaxis().set_visible(False)

plt.subplot(2, 2, 3)
plt.title("Modo de falla III (Binomial)")
plt.plot(x, y3)
plt.xlabel("t")
plt.ylabel("Probabilidad")

plt.subplot(2, 2, 4)
plt.title("Distribucion Combinada")
plt.plot(x, y4)
plt.xlabel("t")

plt.show()

```

Código 3.7: Cálculo de la cdf de un equipo sometido a 3 modos de fallas

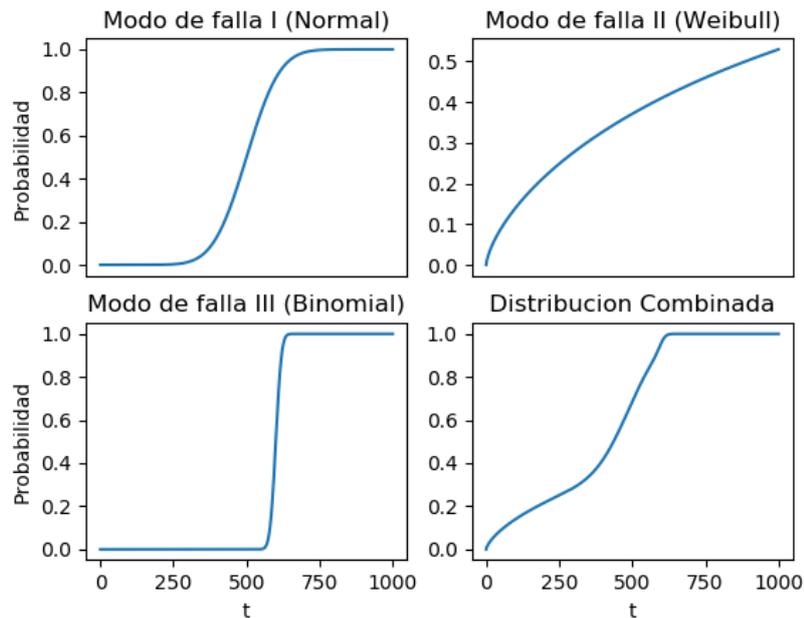


Figura 3.17. Modelos de cdf para 3 modos de falla de un componente

■

El cálculo del MTBF

El MTBF es definido como:

$$MTBF = \int_0^{\infty} R(t) dt \quad (3.12)$$

La Ec. (3.12) puede ser reescrita como:

$$MTBF = \int_0^{\infty} t f(t) dt = \lim_{s \rightarrow 0} R(s) \quad (3.13)$$

Donde s es la transformada de Laplace de la variable t y $R(s)$ es la transformada de Laplace de la función de confiabilidad $R(t)$.

Modelos en serie

Como hemos visto en la Sección 3.1.3, cuando los modos de falla son mutuamente independientes, no hay influencia de uno sobre el otro, pero si una competencia para que uno ocurra primero.

Los modelos en serie siguen el mismo razonamiento, una vez que equipos con bajo o ningún acoplamiento, por ejemplo, en teoría no se influyen a punto de causar una falla uno en el otro. Pero la falla de uno interrumpe el funcionamiento del sistema como un todo.

Dos subsistemas distintos de un sistema que los contiene, pero que en un determinado modelo no se influyen y la falla de cualquiera de ellos interrumpe el funcionamiento del sistema, pueden ser modelados como un sistema en serie. En este caso, tendríamos como ejemplo el sistema de suspensión y el sistema de propulsión (motor y caja de cambio) de un vehículo, cuyo daño en uno no amenaza la confiabilidad del otro, por lo menos en caso de daños poco críticos.

Pero la parada de un motor, o el rompimiento de un resorte de suspensión, impide que el vehículo prosiga en marcha.

La Fig. 3.18, disponible en [NIST/SEMATECH (2012)], presenta un esquema de un sistema compuesto por varios subsistemas en serie, cuyas confiabilidades son independientes.

Sistema en serie reducido a un componente equivalente de sistema

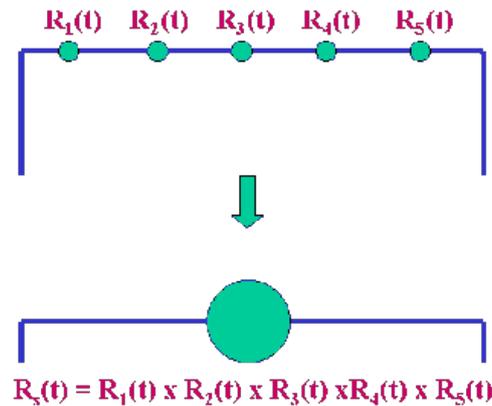


Figura 3.18. Confiabilidad de un sistema compuesto por subsistemas en serie. Copyright 2012 por National Institute of Standards and Technology

Las mismas ecuaciones (3.9), (3.10) y (3.11) permiten el cálculo de la confiabilidad de un sistema en serie.

La duración esperada, hasta la falla, del sistema en serie es:

$$\mathbb{E}_s(t) = \int_0^{\infty} R_s(t) dt \quad (3.14)$$

Modelos en paralelo

Los modelos en paralelo típicos son los sistemas en redundancia sin espera, es decir, sistemas en que uno u otro componente redundante pueden funcionar, pero la elección de cuál componente está en uso, ocurre al azar. No hay una preferencia por uno u otro componente, como ocurre en los casos de aquellos que están en espera (*standby*).

La Ec. (3.15) permite calcular la cdf de un sistema en paralelo. Las otras ecuaciones de interés, para el cálculo de la confiabilidad y de la función de riesgo pueden ser derivadas de la Ec. (3.15) (el índice s indica que la ecuación se refiere al sistema).

$$F_s(t) = \prod_{i=1}^k F_i(t) \quad (3.15)$$

Según [DMAvEx (2010)], la confiabilidad del sistema con k componentes en paralelo es dada por la Ec. (3.16) (modificada).

$$R_s(t) = 1 - F_s(t) = 1 - \prod_{i=1}^k [1 - R_i(t)] \quad (3.16)$$

Entonces:

$$R_s(t) = 1 - \int_0^t f(t) dt = \int_t^{\infty} f(t) dt = e^{-\int_0^t \lambda(t) dt} \quad (3.17)$$

La duración, hasta la falla, del sistema en paralelo está dada por la misma Ec. (3.14), ya fue presentada en otra configuración matemática en la Sección 2.2.

La Fig. 3.19, disponible en [NIST/SEMATECH (2012)], presenta un esquema de un sistema compuesto por varios subsistemas en paralelo, cuyas confiabilidades son independientes.

Sistema en paralelo y componente simple equivalente

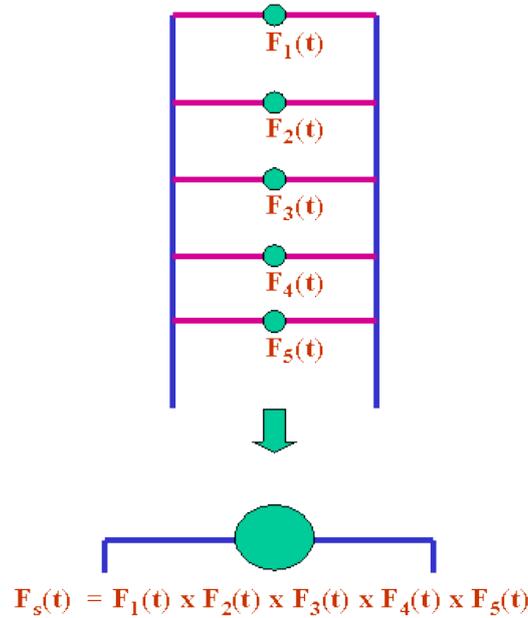


Figura 3.19. Confiabilidad de un sistema compuesto por subsistemas en paralelo. Copyright 2012 por National Institute of Standards and Technology

Modelos en espera (*standby*)

Los sistemas en espera son compuestos por subsistemas redundantes que solamente son accionados mediante la falla de un elemento del sistema. Es decir, un subsistema funcionará exclusivamente hasta su falla, lo que dará inicio al funcionamiento de su repuesto y así sucesivamente.

La Fig. 3.20, disponible en [Dhillon (2002)], presenta un esquema de un sistema compuesto por varios subsistemas en *standby*, cuyas confiabilidades son independientes.

Las siguientes hipótesis serán asumidas para el sistema en *standby*:

- El sistema es compuesto por $k + 1$ subsistemas: 1 en servicio y k en espera;
- El cambio entre los subsistemas es perfecto;
- Todos los subsistemas son independientes e iguales;
- Los subsistemas permanecen tan buenos como nuevos, cuando están en *standby*; y
- La tasa de fallas no es necesariamente constante.

La Ec. (3.18) permite calcular la confiabilidad de un sistema en *standby*.

$$R_s(t) = \sum_{i=0}^k \frac{[\int_0^t \lambda(t) dt]^i e^{-\int_0^t \lambda(t) dt}}{i!} \quad (3.18)$$

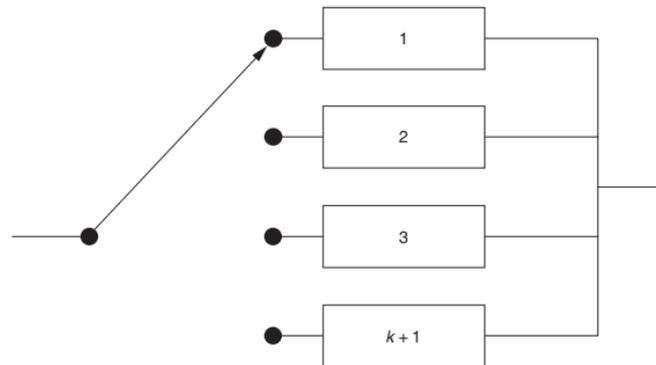


Figura 3.20. Confiabilidad de un sistema compuesto por subsistemas en espera. Copyright 2002 por CRC Press

Si la tasa de fallas es constante, $\lambda(t) = \lambda$, entonces la Ec. (3.18) se queda:

$$R_s(t) = \sum_{i=0}^k \frac{(\lambda t)^i e^{-\lambda t}}{i!} \quad (3.19)$$

En consecuencia de la Ec. (3.19), el MTBF se queda:

$$MTBF = \frac{k+1}{\lambda} \quad (3.20)$$

■ **Ejemplo 3.7** Sea un sistema compuesto por dos subsistemas independientes e idénticos, trabajando en modo *standby*. La tasa de fallas de cada subsistema es de 0,0005 fallas por hora. El Código 3.8 calculará el MTBF y la confiabilidad del sistema para 100 horas de funcionamiento.

```

# -*- coding: utf-8 -*-
import numpy as np
# %%
lamb=0.0005
t=100
n=2
k=n-1
def conf(lamb, t, n):
    r=0
    for i in range(n):
        r=r+((lamb*t)**i)*np.exp(-lamb*t)/np.math.factorial(i)
    return r
def mtbf(k, lamb):
    return (k+1)/lamb
rs=conf(lamb, t, n)
mtbfs=mtbf(k, lamb)
print("La confiabilidad del sistema es R(t)=%.4f" %rs)
print("El MTBF del sistema es MTBF=%d horas" %mtbfs)

```

Código 3.8: Cálculo de MTBF y confiabilidad para un sistema en *standby*

Las salidas del código fueron:

- La confiabilidad del sistema es $R(t)=0.9988$
- El MTBF del sistema es $MTBF=4000$ horas

■



4. Tipos de Mantenimiento

El mantenimiento puede ser definido como las técnicas necesarias para conservar o mantener el funcionamiento adecuado y seguro de sistemas. Según [DECEX (2017)], el objetivo del mantenimiento es evitar la indisponibilidad de equipos, desde su apariencia externa hasta las pérdidas de desempeño.

[Kinnison y Siddiqui (2013)] presentan un abordaje termodinámico al mantenimiento. Considerando que todo sistema termodinámico es sometido a generación de entropía – que siempre crece – algunos equipos se deterioran por el uso y otros por la falta de uso. Esa degradación representa un aumento en la entropía del sistema. En cuanto el ingeniero intenta minimizar la entropía del sistema durante su diseño, la misión del mecánico es combatir el crecimiento natural y continuo de la entropía del sistema durante su ciclo de vida.

La Fig. 4.1 ilustra la tarea del mantenimiento: restaurar el sistema al nivel de perfección conforme sus condiciones de diseño, o sea, reducir la entropía del sistema a sus niveles originales.

4.1 Mantenimiento correctivo

El mantenimiento correctivo es destinado a la reparación o recuperación del material dañado para lo reponer en su condición de uso [DECEX (2017)]. Es realizado después de la ocurrencia de la falla. Es un tipo de mantenimiento que asume que la falla ocurre de modo igualmente probable en cualquier parte, componente o sistema [NASA (2000)]. Eso implica en identificar un grupo específico de repuestos que son más demandados que otros.

Aunque el mantenimiento correctivo sea barato por no demandar planificación o métodos sofisticados de inspección, cuando solamente ese tipo de mantenimiento es practicado suelen ocurrir un alto nivel de actividades de mantenimiento no planificado, elevados stocks de repuestos y uso ineficiente de esfuerzos de mantenimiento, lo que eleva los costos de mantenimiento a lo largo del tiempo.

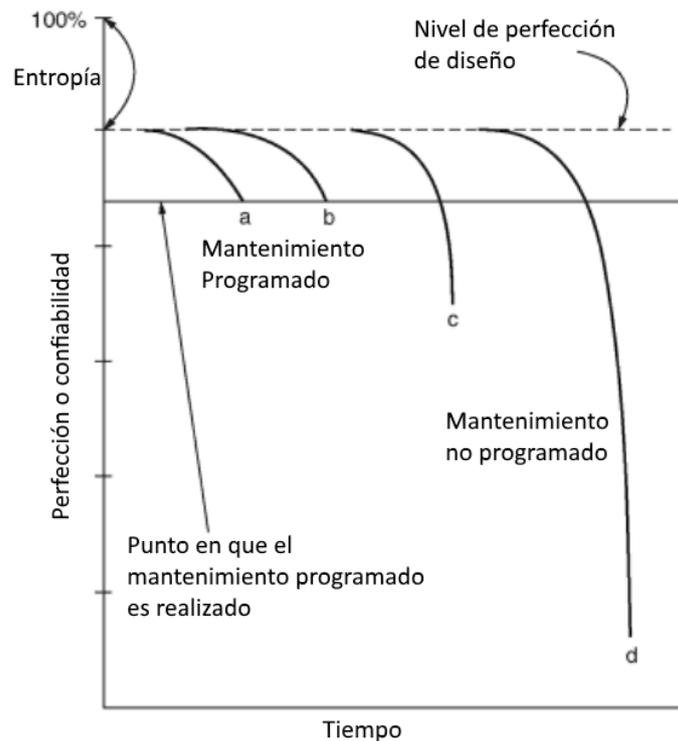


Figura 4.1. Mantenimiento, según la entropía. Copyright 2013 por Mc Graw Hill Education

4.1.1 Medidas de mantenimiento correctivo

Tiempo medio de mantenimiento correctivo

Según [Dhillon (2002)], basado en experiencias técnicas, el mantenimiento correctivo es bien modelado por distribuciones de probabilidades exponencial, normal y lognormal. El autor también cita el ejemplo de equipos electrónicos – los cuales poseen buena capacidad de prueba y rápida remoción, y sustitución de componentes – que generalmente son bien modelados por la distribución exponencial. Para componentes mecánicos y electromecánicos, la distribución normal es frecuentemente empleada.

La Ec. (4.1) define el tiempo medio de mantenimiento correctivo.

$$T_{mmc} = \frac{\sum \lambda_j T_{mcj}}{\sum \lambda_j} \quad (4.1)$$

Donde:

- T_{mmc} - tiempo medio de mantenimiento correctivo;
- T_{mcj} - tiempo de mantenimiento correctivo para el j-ésimo elemento del equipo o sistema;
- λ_j - tasa de falla del j-ésimo elemento del equipo o sistema.

Tiempo mediano activo de mantenimiento correctivo

Es una medida del tiempo según la cual 50% de todo mantenimiento correctivo puede ser cumplido. Es una ecuación dependiente del tipo de distribución que representa el tiempo de mantenimiento correctivo. Si la distribución representativa es la lognormal, la Ec. (4.2), donde

T_{med} es el tiempo mediano activo, permite calcular ese tiempo.

$$T_{med} = \text{antilog} \left(\frac{\sum \lambda_j \log T_{mcj}}{\sum \lambda_j} \right) \quad (4.2)$$

Tiempo máximo activo de mantenimiento correctivo

Es una medida del tiempo necesario para cumplir con 90 o 95 % de todas las acciones de mantenimiento correctivo.

Si la distribución representativa es la lognormal, la Ec. (4.3) permite calcular el tiempo máximo de mantenimiento.

$$T_{max} = \text{antilog}(T_{mn} + z\sigma_{cm}) \quad (4.3)$$

Donde:

- T_{max} - tiempo máximo activo de mantenimiento correctivo;
- T_{mn} - media de los logaritmos de T_{cmj} ;
- σ_{cm} - desviación estándar de los logaritmos de la muestra de los tiempos de mantenimiento correctivo;
- z - desviación estándar correspondiente al valor percentil especificado para T_{max} .

El valor de σ_{cm} es calculado por la Ec. (4.4), donde M es el número total de tiempos de mantenimiento correctivo:

$$\sigma_{cm} = \left[\frac{\sum_{j=1}^M (\log T_{cmj})^2 - \frac{(\sum_{j=1}^M \log T_{cmj})^2}{M}}{M-1} \right]^{\frac{1}{2}} \quad (4.4)$$

4.1.2 Modelos matemáticos de mantenimiento correctivo

Para la mejor comprensión de los modelos a seguir, serán presentados los conceptos de Disponibilidad (A), conforme definidos en [NASA (2000)].

1. Disponibilidad inherente (A_i)
 - a. La probabilidad de un sistema o equipo, cuando es usado bajo determinadas condiciones en un ambiente de soporte ideal, opera satisfactoriamente en cualquier punto en el tiempo, como se requiere;
 - b. Excluye acciones de mantenimiento preventivo o planificado por el tiempo, retrasos logísticos o administrativos;
 - c. Se calcula por la Ec. (4.5).
2. Disponibilidad alcanzada (A_a)
 - a. La probabilidad de un sistema o equipo, cuando es usado bajo determinadas condiciones en un ambiente de soporte ideal, opera satisfactoriamente a cualquier punto en el tiempo;
 - b. Mantenimiento preventivo está incluido;
 - c. Excluye retrasos logísticos o administrativos.
3. Disponibilidad operacional (A_o)
 - a. La probabilidad de un sistema o equipo, cuando es usado bajo determinadas condiciones en un ambiente de soporte ideal, opera satisfactoriamente a algún punto en el tiempo, cuando se necesite;

- b. Incluye tiempo de mantenimiento activo (que será definido en la Sección 4.2.1), retrasos logísticos o administrativos;
- c. Se calcula por la Ec. (4.6), donde MLDT es el tiempo medio de retrasos logísticos.

$$A_i = \frac{MTBF}{MTBF+MTTR} \quad (4.5)$$

$$A_o = \frac{MTBF}{MTBF+MTTR+MLDT} \quad (4.6)$$

Los modelos y ejemplos a continuación siguen la metodología presentada por [Dhillon (2002)].

Modelo I

Este modelo matemático, basado en un diagrama de transición entre estados, representa un sistema que está o no operando.

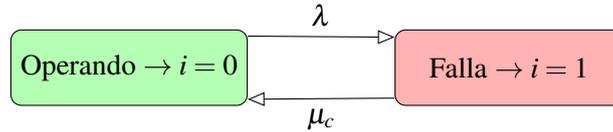


Figura 4.2. Diagrama de transición de sistema

Para este modelo, serán adoptadas las siguientes premisas:

- La tasa de fallas y las tasas de mantenimiento correctivo son constantes;
- El sistema arreglado es tan bueno cuanto nuevo;
- Las fallas del sistema son estadísticamente independientes.

Sean las variables:

- i - estado del sistema;
- $P_i(t)$ - probabilidad del estado i a un cierto tiempo t ;
- λ - tasa de fallas;
- μ_c - tasa de mantenimiento correctivo del sistema.

Asumiéndose el desarrollo del modelo basado en cadenas de Markov [Dhillon (2002)], la disponibilidad del sistema está dada por la Ec. (4.7).

$$A_s(t) = P_0(t) = \frac{\mu_c}{\lambda + \mu_c} + \frac{\lambda}{\lambda + \mu_c} e^{-(\lambda + \mu_c)t} \quad (4.7)$$

La probabilidad del sistema encontrarse en Modo I (falla) es:

$$P_1(t) = \frac{\mu_c}{\lambda + \mu_c} - \frac{\lambda}{\lambda + \mu_c} e^{-(\lambda + \mu_c)t} \quad (4.8)$$

Considerándose $\lambda = 1/MTBF$, $\mu_c = 1/T_{mmc}$, y que el sistema está en régimen permanente ($t \rightarrow \infty$) la Ec. (4.7) se queda:

$$A_s(t) = \frac{MTBF}{T_{mmc} + MTBF} \quad (4.9)$$

■ **Ejemplo 4.1** Sea un componente de un equipo que posee un MTBF=3000 h y su $T_{mmc} = 5 h$. Si el sistema alcanzó el régimen permanente y las fallas del equipo, bien como sus tiempos de mantenimiento correctivo, son bien modelados por una distribución exponencial, la disponibilidad del equipo se queda:

$$A_s(t) = \frac{3000}{5 + 3000} = 0,9983 \quad (4.10)$$

■

Modelo II

Este modelo representa un sistema que puede fallar por dos modos (modo I y modo II). Según [Dhillon (2002)], un caso típico puede ser observado en una válvula de flujo, que puede fallar estando abierta o cerrada.

En ambos los casos, el mantenimiento correctivo pone el equipo en su estado operativo.

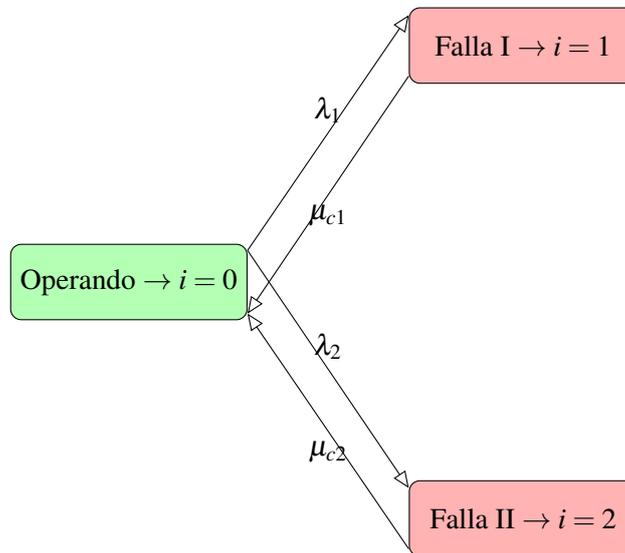


Figura 4.3. Diagrama de transición del sistema

Para este modelo, serán adoptadas las siguientes premisas:

- El sistema puede fallar de dos modos mutuamente independientes;
- La tasa de fallas y las tasas de mantenimiento correctivo son constantes;
- El sistema arreglado es tan bueno cuanto nuevo;
- Las fallas del sistema son estadísticamente independientes.

Las probabilidades del sistema para encontrarse en uno de los tres estados son dadas por las siguientes ecuaciones:

$$P_0(t) = \frac{\mu_{c1}\mu_{c2}}{m_1m_2} + \left[\frac{(m_1+\mu_{c1})(m_1+\mu_{c2})}{m_1(m_1-m_2)} \right] e^{m_1t} - \left[\frac{(m_2+\mu_{c1})(m_2+\mu_{c2})}{m_2(m_1-m_2)} \right] e^{m_2t} \quad (4.11)$$

$$P_1(t) = \frac{\lambda_1\mu_{c2}}{m_1m_2} + \left[\frac{(\lambda_1m_1+\lambda_1\mu_{c2})}{m_1(m_1-m_2)} \right] e^{m_1t} - \left[\frac{(\mu_{c1}+m_2)\lambda_1}{m_2(m_1-m_2)} \right] e^{m_2t} \quad (4.12)$$

$$P_2(t) = \frac{\lambda_2 \mu_{c1}}{m_1 m_2} + \left[\frac{(\lambda_2 m_1 + \lambda_2 \mu_{c1})}{m_1 (m_1 - m_2)} \right] e^{m_1 t} - \left[\frac{(\mu_{c1} + m_2) \lambda_2}{m_2 (m_1 - m_2)} \right] e^{m_2 t} \quad (4.13)$$

Donde:

$$m_1, m_2 = \frac{-A \pm (A^2 - 4B)^{\frac{1}{2}}}{2} \quad (4.14)$$

$$A = \mu_{c1} + \mu_{c2} + \lambda_1 + \lambda_2 = -(m_1 + m_2) \quad (4.15)$$

$$B = \mu_{c1} \mu_{c2} + \lambda_1 \mu_{c2} + \lambda_2 \mu_{c1} = m_1 m_2 \quad (4.16)$$

La disponibilidad del sistema, considerando el régimen en estado permanente $A_s(t) = P_0(t) \forall t \rightarrow \infty$ está dada por la Ec. (4.17):

$$A_s(t) = \frac{\mu_{c1} \mu_{c2}}{m_1 m_2} = \frac{\mu_{c1} \mu_{c2}}{\mu_{c1} \mu_{c2} + \lambda_1 \mu_{c2} + \lambda_2 \mu_{c1}} \quad (4.17)$$

■ **Ejemplo 4.2** Sea un sistema de ingeniería que puede fallar de dos modos mutuamente independientes. Las tasas de fallas son $\lambda_1 = 0,002$ fallas/h y $\lambda_2 = 0,005$ fallas/h. Las tasas de mantenimiento correctivo son $\mu_{c1} = 0,006$ arreglos/h y $\mu_{c2} = 0,009$ arreglos/h. Considerando un tiempo muy largo, la disponibilidad es:

$$A_s(t) = \frac{0,006 * 0,009}{(0,006 * 0,009) + (0,002 * 0,009) + (0,005 * 0,006)} = 0,5294 \quad (4.18)$$

■

Modelo III

En este modelo, es posible al sistema operar en modo degradado. [Dhillon (2002)] ejemplifica con un generador operando en modo degradado, que produce electricidad a una potencia intermedia.

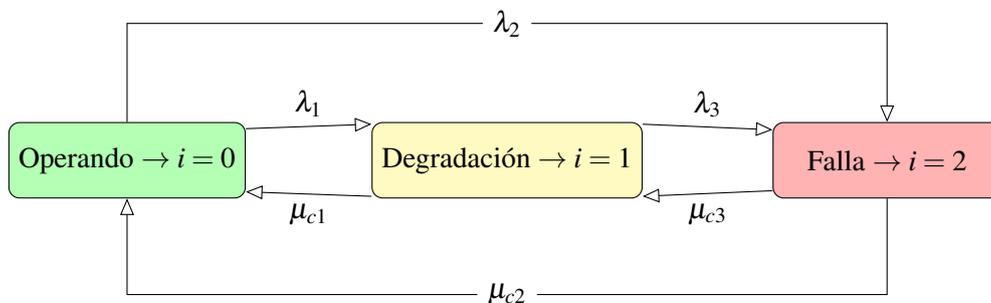


Figura 4.4. Diagrama de transición de sistema

Para este modelo, serán adoptadas las siguientes premisas:

- El sistema puede fallar completamente o parcialmente y las tasas de mantenimiento correctivo son constantes;
- El sistema puede operar totalmente o parcialmente. La operación parcial del sistema puede parar el sistema completamente;
- El sistema arreglado es tan bueno cuanto nuevo;
- Las fallas del sistema son estadísticamente independientes.

Las siguientes ecuaciones modelan la probabilidad de fallas del sistema [Dhillon (2002)].

$$\begin{aligned}
 P_0(t) &= P1 + P2 + (1 - P1 - P3)e^{K_2 t} \\
 P1 &= \frac{\mu_{c1}\mu_{c2} + \lambda_3\mu_{c2} + \mu_{c1}\mu_{c3}}{K_1 K_2} \\
 P2 &= \left[\frac{\mu_{c1}K_1 + \mu_{c2}K_2 + \mu_{c3}K_1 + K_1\lambda_3 + K_1^2 + \mu_{c1}\mu_{c2} + \lambda_3\mu_{c2} + \mu_{c1}\mu_{c3}}{Y} \right] e^{K_1 t} \\
 P3 &= \left[\frac{\mu_{c1}K_1 + \mu_{c2}K_1 + \mu_{c3}K_1 + K_1\lambda_3 + K_2^2 + \mu_{c1}\mu_{c2} + \lambda_3\mu_{c2} + \mu_{c1}\mu_{c3}}{Y} \right]
 \end{aligned} \tag{4.19}$$

Donde:

$$Y = K_1(K_1 - K_2) \tag{4.20}$$

$$\begin{aligned}
 P_1(t) &= P1 + P2e^{K_1 t} - (P1 + P2)e^{K_2 t} \\
 P1 &= \frac{\lambda_1\mu_{c2} + \lambda_1\mu_{c3} + \lambda_2\mu_{c3}}{K_1 K_2} \\
 P2 &= \frac{K_1\lambda_1 + \lambda_1\mu_{c2} + \lambda_1\mu_{c3} + \lambda_2\mu_{c3}}{Y}
 \end{aligned} \tag{4.21}$$

$$\begin{aligned}
 P_2(t) &= P1 + P2e^{K_1 t} - (P1 + P2)e^{K_2 t} \\
 P1 &= \frac{\lambda_1\lambda_3 + \lambda_2\mu_{c1} + \lambda_2\lambda_3}{K_1 K_2} \\
 P2 &= \frac{K_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\mu_{c1} + \lambda_2\lambda_3}{Y}
 \end{aligned} \tag{4.22}$$

Donde:

$$K_1, K_2 = \frac{-D \pm (D^2 - 4F)^{1/2}}{2} \tag{4.23}$$

$$D = \mu_{c1} + \mu_{c2} + \mu_{c3} + \lambda_1 + \lambda_2 + \lambda_3 \tag{4.24}$$

$$F = K_1 K_2 = \mu_{c1}\mu_{c2} + \lambda_3\mu_{c2} + \mu_{c1}\mu_{c3} + \lambda_1\mu_{c2} + \lambda_1\mu_{c3} + \lambda_1\lambda_3 + \lambda_2\mu_{c1} + \lambda_2\mu_{c3} + \lambda_2\lambda_3 \tag{4.25}$$

Para evaluar la disponibilidad del sistema en operación total o parcial, la disponibilidad debe considerar la suma de las dos respectivas probabilidades, o sea: $A_s(t) = P_0(t) + P_1(t)$. Así, cuando $t \rightarrow \infty$, las disponibilidad se queda:

$$A_s = \frac{\mu_{c1}\mu_{c2} + \lambda_3\mu_{c2} + \mu_{c1}\mu_{c3} + \lambda_1\mu_{c2} + \lambda_1\mu_{c3} + \lambda_2\mu_{c3}}{K_1 K_2} \tag{4.26}$$

En el caso de disponibilidad total, tenemos:

$$A_s = P_0 = \frac{\mu_{c1}\mu_{c2} + \lambda_3\mu_{c2} + \mu_{c1}\mu_{c3}}{K_1 K_2} \tag{4.27}$$

■ **Ejemplo 4.3** El Código 4.1 presentará el cálculo de un sistema en funcionamiento pleno, en estado permanente. Para ese sistema, son dados los siguientes parámetros: $\lambda_1 = 0,002$ fallas/h, $\lambda_2 = 0,003$ fallas/h, $\lambda_3 = 0,001$ fallas/h, $\mu_{c1} = 0,006$ arreglos/h, $\mu_{c2} = 0,004$ y $\mu_{c3} = 0,008$ arreglos/h.

```
# -*- coding: utf-8 -*-
lamb1=0.002
lamb2=0.003
lamb3=0.001
mu1=0.006
mu2=0.004
mu3=0.008
k1k2=mu1*mu2+lamb3*mu2+mu1*mu3+mu2*lamb1+lamb1*mu3+lamb1*lamb3+mu1*lamb2+
      lamb2*mu3+lamb2*lamb3
asf=(mu1*mu2+lamb3*mu2+mu1*mu3)/k1k2
print("Hay aproximadamente %.2f%%de probabilidad que el sistema permanezca
      en produccion total" %(asf*100))
```

Código 4.1: Cálculo de disponibilidad para un sistema con posibilidad de operación parcial

La salida del código fue: “Hay aproximadamente 51.70% de probabilidad que el sistema permanezca en producción total”.

■

Modelo IV

En este modelo, el sistema presenta una redundancia de dos unidades iguales. El mantenimiento correctivo es aplicado inmediatamente cuando ocurre una falla de una unidad.



Figura 4.5. Diagrama de transición de sistema

Para este modelo, serán adoptadas las siguientes premisas:

- El sistema está compuesto por dos unidades independientes e iguales;
- No se aplica mantenimiento correctivo cuando las dos unidades fallan;
- El sistema arreglado es tan bueno cuanto nuevo;
- Las tasas de fallas y de mantenimiento son constantes.

Las siguientes ecuaciones modelan la probabilidad de fallas del sistema [Dhillon (2002)].

$$P_0(t) = \left[\frac{\lambda + \mu + C_1}{C_1 - C_2} \right] e^{C_1 t} - \left[\frac{\lambda + \mu + C_1}{C_1 - C_2} \right] e^{C_2 t} \quad (4.28)$$

$$P_1(t) = \left[\frac{2\lambda}{C_1 - C_2} \right] e^{C_1 t} - \left[\frac{2\lambda}{C_1 - C_2} \right] e^{C_2 t} \quad (4.29)$$

$$P_2(t) = 1 + \left[\frac{C_2}{C_1 - C_2} \right] e^{C_1 t} - \left[\frac{C_1}{C_1 - C_2} \right] e^{C_2 t} \quad (4.30)$$

Donde:

$$C_1, C_2 = \frac{[-(3\lambda + \mu) \pm \sqrt{(3\lambda + \mu)^2 - 8\lambda^2}]^{\frac{1}{2}}}{2} \quad (4.31)$$

$$C_1 C_2 = 2\lambda^2 \quad (4.32)$$

$$C_1 + C_2 = -(3\lambda + \mu) \quad (4.33)$$

La disponibilidad del sistema está dada por la siguiente ecuación, que es la probabilidad del sistema para estar en el estado 0 o en el estado 1:

$$R_s(t) = P_0(t) + P_1(t) \quad (4.34)$$

Así, el MTBF para sistemas con una redundancia puede ser calculado por la siguiente ecuación, donde μ_{Cm} es la tasa de mantenimiento correctivo de una unidad redundante:

$$MTBF_s = \int_0^{\infty} R_s(t) dt = \frac{3\lambda + \mu_{Cm}}{2\lambda^2} \quad (4.35)$$

Asumiéndose $\lambda = 1/MTBF_u$ y $\mu_{Cm} = 1/MCMT$, donde MCMT es la abreviatura para tiempo medio para el mantenimiento correctivo de una unidad, entonces:

$$MTBF_s = \frac{MTBF_u}{2MCMT} (3MCMT + MTBF_u) \quad (4.36)$$

■ **Ejemplo 4.4** Considere un sistema que posee dos subsistemas en paralelo. En este sistema, es posible arreglar la falla de un subsistema en cuanto el otro soporta las operaciones, pero la falla de ambos subsistemas (es decir, la falla del sistema) no permite el arreglo. Los parámetros del sistema son: $MTBF_u = 150 h$ y $MCMT = 5 h$. Los cálculos abajo suministrarán el MTBF con y sin soporte de mantenimiento correctivo, respectivamente.

Desde la Ec. (4.36):

$$MTBF_s = \frac{150}{2 * 5} (3 * 5 + 150) = 2475 h \quad (4.37)$$

Asumiéndose $\mu_{Cm} = 0$ en la Ec. (4.35), o sea, si falla un subsistema, el otro asume inmediatamente hasta su falla, sin arreglo del primer subsistema que se quedó inoperante:

$$MTBF_s = \frac{3}{\lambda} = 225 h \quad (4.38)$$

■

Sistemas con redundancia – casos específicos

En esta sección, serán presentadas dos soluciones aproximadas para el cálculo de la tasa de fallas efectiva de sistemas con redundancia [Dhillon (2002)]. Es importante recordar que la tasa de fallas efectiva es el recíproco del MTBF del subsistema o del sistema.

1. Múltiples redundancias idénticas, donde k redundancias pueden fallar sin perjuicio del funcionamiento del sistema.

Sean m redundancias independientes e idénticas. La Ec. (4.39) permite el cálculo de la tasa de fallas del sistema, considerando el sistema operativo con k unidades inoperantes ($m > k$). Para que el sistema opere, por lo menos $(m-k)$ subsistemas deberán permanecer en operación.

$$\lambda_{(m-k)/m} = \frac{m! \lambda^{k+1}}{(m-k-1)! \mu^k} \quad (4.39)$$

■ **Ejemplo 4.5** Considere un sistema que posee tres subsistemas independientes e idénticos en paralelo. Al menos, dos subsistemas necesitan operar normalmente para que el sistema opere correctamente. Asumiéndose la tasa de falla como $\lambda = 0,0001$ fallas por hora y en media 2 horas de arreglo por unidad, siguiendo una distribución exponencial, se desea calcular la tasa de falla efectiva si no hay reparaciones previstas.

Desde la Ec. (4.39):

$$\lambda_{(3-1)/3} = \frac{3! \lambda^2}{1! \mu} = 1,2 * 10^{-7} \text{ fph} \quad (4.40)$$

■

2. Dos redundancias distintas y en paralelo, donde el mantenimiento empieza luego que cualquier una de las redundancias falla.

La Ec. (4.41) permite el cálculo de la tasa de fallas del sistema, considerando que las tasas de falla y las tasas de mantenimiento correctivo son constantes.

$$\lambda_{se} = \frac{\lambda_1 \lambda_2 [(\mu_1 + \mu_2) + (\lambda_1 + \lambda_2)]}{\mu_1 \mu_2 + (\mu_1 + \mu_2) (\lambda_1 + \lambda_2)} \quad (4.41)$$

■ **Ejemplo 4.6** Considere un sistema que posee dos subsistemas independientes y no idénticos en paralelo. Los sistemas son caracterizados por los siguientes parámetros: $\lambda_1 = 0,004$ fph, $\mu_1 = 0,005$ arreglos por hora, $\lambda_2 = 0,002$ fph, $\mu_2 = 0,003$ arreglos por hora. Se desea calcular la tasa de falla efectiva.

Desde la Ec. (4.41):

```
# -*- coding: utf-8 -*-      1
lamb1=0.004                  2
mu1=0.005                   3
lamb2=0.002                  4
mu2=0.003                   5
lambsis=(lamb1*lamb2*((mu1+mu2)+(lamb1+lamb2)))/(mu1*mu2+(mu1+mu2)*(lamb1+  6
    lamb2))
print("Se estima una tasa de fallas de %.4f fph" %lambsis)      7
```

Código 4.2: Cálculo de tasa de fallas para un sistema con redundancias distintas

La salida del código fue: “Se estima una tasa de fallas de 0,0018 fph”.

■

3. Dos redundancias iguales y en paralelo, donde las dos fallan por una causa común.

En este modelo, el sistema puede fallar por causas normales, pero también las dos redundancias pueden fallar por causas simultáneas.

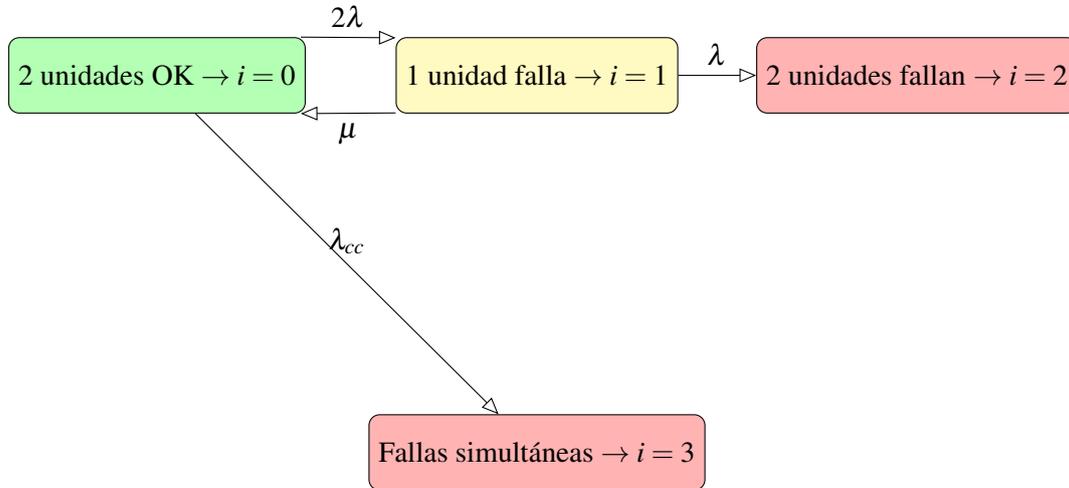


Figura 4.6. Diagrama de transición de sistema

Para el modelaje de este sistema por medio del método de Markov, será aplicada una transformada de Laplace a las ecuaciones ya conocidas. Las probabilidades de falla serán modeladas en función de la disponibilidad.

$$P_0(s) = \frac{s+\lambda+\mu}{A} \quad (4.42)$$

La disponibilidad se queda:

$$A = (s + 2\lambda + \lambda_{cc})(s + \lambda + \mu) - 2\lambda\mu \quad (4.43)$$

Entonces:

$$P_1(s) = \frac{2\lambda}{A} \quad (4.44)$$

$$P_2(s) = \frac{2\lambda^2}{sA} \quad (4.45)$$

$$P_3(s) = \frac{\lambda_{cc}(s+\lambda+\mu)}{sA} \quad (4.46)$$

La transformada de Laplace de la confiabilidad se queda:

$$R_{ps}(s) = P_0(s) + P_1(s) = \frac{s+3\lambda+\mu}{A} \quad (4.47)$$

El MTBF se queda:

$$MTBF_{ps} = \lim_{s \rightarrow 0} R_{ps}(s) = \frac{3\lambda + \mu}{2\lambda^2 + \lambda\lambda_{cc} + \lambda_{cc}\mu} \quad (4.48)$$

4.2 Mantenimiento preventivo

El mantenimiento preventivo es un procedimiento periódico, que prevé un conjunto de inspecciones, ajustes, limpiezas, lubricaciones, sustitución de componentes, calibraciones y reparos de componentes y equipos.

Ese tipo de mantenimiento puede ser basado en el tiempo de funcionamiento efectivo del equipamiento (pueden ser incluidas las cantidades de ciclos de utilización, como aterrizajes y despegues de aeronaves, por ejemplo), y/o en intervalos determinados, aunque en este caso el equipo no haya sido utilizado.

Para la determinación de los planes de mantenimiento preventivo, es necesario que se conozcan los patrones estadísticos de las fallas, sus intervalos de ocurrencias y sus criticidades. Hay fallas que no ocurren con alta frecuencia, pero son críticas, y fallas que ocurren con frecuencia más elevada, pero son de fácil arreglo y bajo costo. En general, el mantenimiento preventivo también está dedicado a las piezas y componentes que sufren desgaste, cuya deterioración sigue una condición de razonable previsibilidad.

Otro aspecto a ser considerado en el mantenimiento preventivo, es que su ejecución trae beneficios al sistema, es decir, una sustitución de componente para volver el sistema a la condición de nuevo no introduce nuevos daños o desgastes de otros componentes. En el caso de que suceda, esos componentes también son sustituidos por repuestos.

Una vez que en general el mantenimiento preventivo se basa en MTBF, y debido al hecho que el MTBF es una medida promedia, los daños al azar pueden ocurrir más temprano o más tarde que el previsto. Si ocurren más tarde, el mantenimiento preventivo puede sustituir componentes todavía buenos, innecesariamente. Dependiendo del costo de mantenimiento de ciertos componentes o sistemas, otros tipos y técnicas de mantenimiento más avanzados deben ser empleados.

Una vez más es importante destacar que los planes de mantenimiento preventivo funcionan cuando los sistemas son sometidos a las condiciones de funcionamiento para los cuales fueron diseñados y estudiados. Condiciones de empleo más rigurosas pueden reducir los tiempos de falla, impactando negativamente en la previsibilidad de fallas conocidas del sistema.

Según [DECEX (2017)], con el mantenimiento preventivo la frecuencia de fallas tiende a disminuir, lo que aumenta la disponibilidad de los equipos y, consecuentemente, disminuye las paradas inoportunas y los costos. Además de eso, es posible conseguir una mejor optimización de los stocks de repuestos, identificando la demanda necesaria y disminuyendo los costos de almacenamiento, empleo y mantenimiento de stocks.

Si uno de los principales retos del mantenimiento es disminuir costos, el mantenimiento preventivo debe ser ajustado para que sea realizado a intervalos óptimos, para que no se queden demasiado onerosos. En ese sentido, [Dhillon (2002)] presenta la siguiente fórmula para la decisión de proceder a un programa de mantenimiento preventivo:

$$\boxed{(NB)(ACPBD)(\alpha) > CPMS} \quad (4.49)$$

La Ec. (4.49) está compuesta por los siguientes factores:

- CPMS - costo total del sistema de mantenimiento preventivo;
- α - un factor de corrección, que es propuesto como 70 % de los costos de paradas debidas a averías;
- NB - número de paradas debidas a averías;
- ACPBD - costo medio por parada debido a averías.

Siguiendo con la metodología de cálculo presentada por [Dhillon (2002)], las próximas secciones permitirán el cálculo de medidas de mantenimiento preventivo.

Tiempo medio de mantenimiento preventivo (MPMT)

El MPMT es la media de tiempo de parada – ponderada por la frecuencia – necesario para la ejecución de un mantenimiento preventivo. Esa métrica no considera tiempos administrativos y logísticos involucrados con en mantenimiento. Según [DOD (1966)], es expresado en términos de hombres-horas.

$$MPMT = \frac{\sum_{i=1}^m f_i MPMT_i}{\sum_{i=1}^m f_i} \quad (4.50)$$

La Ec. (4.50) está compuesta por los siguientes factores:

- $MPMT_i$ - tiempo necesario para realizar el i -ésimo mantenimiento preventivo;
- f_i - frecuencia de acciones de mantenimiento preventivo.

Tiempo mediano de mantenimiento preventivo (MDPMT)

El MDPMT, así como su similar descrito en la Sección 4.1.1, es el tiempo necesario de parada para conducir 50% de todas las acciones de mantenimiento preventivo. Para tiempos de mantenimiento preventivo modelados como una distribución lognormal, el MDPMT puede ser calculado como:

$$MDPMT = \text{antilog} \left[\frac{\sum_{i=1}^m \lambda_i \log MPMT_i}{\sum_{i=1}^m \lambda_i} \right] \quad (4.51)$$

En la Ec. (4.51), λ_i es la tasa de falla constante del elemento i del equipo donde el mantenimiento está siendo evaluado, ajustado para factores como ciclos de trabajo sin interrupción, fallas de tolerancia e interacciones, y fallas catastróficas.

Cuando la tasa de reparos sigue una distribución normal, la mediana coincide con la media de la distribución y el MDPMT se queda igual al MPMT [DOD (1966)].

Cuando la tasa de reparos sigue una distribución exponencial, entonces $MDPMT = 0,69 * MPMT$ [DOD (1966)].

Cuando la tasa de reparos sigue una distribución lognormal, entonces $MDPMT = \frac{MPMT}{\text{antilog}(1,15\sigma^2)} = 0,45 * MPMT$, donde σ es la desviación estándar de los logaritmos de base 10 de los tiempos de reparo [DOD (1966)].

Tiempo máximo de mantenimiento preventivo (MXPMT)

El MXPMT [Dhillon (2002)], así como su similar descrito en la Sección 4.1.1, es el tiempo máximo de parada necesario para cumplir un porcentual de mantenimiento preventivo. Para distribuciones de tiempos de mantenimiento preventivo lognormales, el MXPMT es calculado por la siguiente ecuación:

$$MXPMT = \text{antilog} (\log MPMT_m + yS_{\log MPMT}) \quad (4.52)$$

El valor de y para distribuciones normales es tablado. Para el percentil 90, $y = 1,283$ y para el percentil 95 $y = 1,645$. El $\log MPMT_m$ es la media de los logaritmos de $MPMT_i$, dado por las siguientes ecuaciones:

$$\log MPMT_m = \frac{\sum_{i=1}^m \lambda_i \log MPMT_i}{\sum_{i=1}^m \lambda_i} \quad (4.53)$$

$$S_{\log MPMT} = \left(\frac{\sum_{i=1}^m (\log MPMT_i)^2 - \frac{(\sum_{i=1}^m \log MPMT_i)^2}{m}}{m-1} \right)^{\frac{1}{2}} \quad (4.54)$$

4.2.1 Modelos matemáticos de mantenimiento preventivo

Modelo I de optimización de inspección

Este modelo tiene por objetivo calcular el número óptimo de inspecciones por instalación de mantenimiento, por unidad de tiempo. Si TDT es el tiempo de paradas por unidad de tiempo destinado a una instalación para hacerse un mantenimiento preventivo, entonces:

$$TDT = yT_i + \frac{cT_b}{y} \quad (4.55)$$

Donde:

- c - constante adoptada para una determinada instalación de mantenimiento;
- T_b - tiempo de parada de una instalación por motivo de un mantenimiento de una avería o falla;
- T_i - tiempo de parada de una instalación por motivo de inspección;
- y - número de inspecciones por instalación, por unidad de tiempo.

Cuando se deriva la Ec. (4.55) en el tiempo y y se iguala a cero, se obtiene el número óptimo de inspecciones por instalación por unidad de tiempo y^* y el tiempo total óptimo de paradas por unidad de tiempo por instalación (TDT^*).

$$y^* = \left(\frac{cT_b}{T_i} \right)^{\frac{1}{2}} \Rightarrow TDT^* = 2(cT_iT_b)^{\frac{1}{2}} \quad (4.56)$$

■ **Ejemplo 4.7** Considere una organización que durante un periodo de tiempo observó las siguientes características de mantenimiento preventivo: $T_b = 0,1$ mes, $T_i = 0,05$ mes y $c=3$.

Desde la Ec. (4.56):

$$y^* = \left(\frac{3 \cdot 0,1}{0,05} \right)^{\frac{1}{2}} = 2,45 \quad (4.57)$$

En este caso, el número óptimo es de 2 inspecciones por mes.

■

Mantenimiento activo

Según [DOD (1966)], es el tiempo medio de mantenimiento en que la suma de mantenimiento preventivo y correctivo requieren, en hombres-hora, para atender a un producto por un tiempo específico, dividido por el número total de mantenimientos preventivos y correctivos durante el mismo tiempo.

$$\bar{M} = \frac{(\sum \lambda) \bar{M}_{ct} + (\sum f) \bar{M}_{pt}}{\sum \lambda_i + \sum f_j} \quad (4.58)$$

Donde:

- \bar{M} - tiempo medio de mantenimiento activo;
- \bar{M}_c - tiempo medio de mantenimiento correctivo (asociado al tiempo t_i);
- \bar{M}_p - tiempo medio de mantenimiento preventivo (asociado al tiempo t_j);
- $\sum \lambda$ - suma de las tasas de fallas de las partes;
- $\sum f$ - suma de la frecuencia de ocurrencias de las actividades de mantenimiento preventivo;
- t_i - tiempo de operación durante el período t_j ;
- t_j - tiempo de calendario en proceso operativo.

Confiabilidad y el modelo de determinación del tiempo medio de falla de un sistema con mantenimiento periódico

Para este modelo matemático, serán asumidas las siguientes premisas:

- Una parte con daño será repuesta por otra parte nueva y estadísticamente idéntica;
- Las actividades de mantenimiento periódico son desarrolladas en el sistema a cada Y horas, empezando al tiempo cero.

El intervalo de tiempo del mantenimiento periódico es calculado por:

$$y = iY + t \quad \forall i = 0, 1, 2, \dots; 0 \leq t < Y \quad (4.59)$$

Según [Dhillon (2002)], para $i=1$ y $T=0$, la confiabilidad de un sistema redundante sometido a un mantenimiento periódico a cada Y horas es dado por:

$$R_Y(y = Y) = R(Y) \quad (4.60)$$

Para $i=2$ y $T=0$, con el sistema operando las primeras Y horas sin sufrir fallas y las otras Y horas sin fallas después de sustituir cualquier parte dañada:

$$R_Y(y = 2Y) = [R(Y)]^2 \quad (4.61)$$

Sumándose otras T horas sin fallas, para $0 < T < Y$, tenemos:

$$R_Y(y = 2Y + T) = [R(Y)]^2 R(T) \quad (4.62)$$

En general:

$$R_Y(y = iY + T) = [R(Y)]^i R(T) \quad \forall i = 0, 1, 2, \dots; 0 \leq t < Y \quad (4.63)$$

Asumiéndose que el $MTBF_{mp} = \int_0^{\infty} R_Y(y) dy$, entonces:

$$MTBF_{mp} = \frac{\int_0^Y R(T) dT}{1 - R(Y)} \quad (4.64)$$

- **Ejemplo 4.8** Considere dos subsistemas independientes e idénticos, en paralelo, de un sistema. Cada subsistema presenta un $MTBF=200$ h, conforme una distribución exponencial. El mantenimiento preventivo es realizado cada 100 h. Se desea calcular el $MTBF$ con y sin la influencia del mantenimiento preventivo.

Desde la Ec. (3.17), para sistemas en paralelo exponencialmente distribuidos:

$$R_s(t) = 1 - \prod_{i=1}^n (1 - e^{-\lambda_i t}) \quad (4.65)$$

Por lo tanto:

$$R_s(t) = 1 - \prod_{i=1}^n (1 - e^{-\lambda_i t}) \Rightarrow R(y) = 2e^{-y/200} - e^{-2y/200} \quad (4.66)$$

Sustituyendo la Ec. (4.66) en la Ec. (4.64):

$$MTBF_{mp} = \frac{\int_0^{100} (2e^{-t/200} - e^{-2t/200}) dt}{1 - (2e^{-100/200} - e^{-2 \cdot 100/200})} = 608,26 \text{ h} \quad (4.67)$$

Para el caso de no haber mantenimiento preventivo, se debe sustituir la Ec. (4.66) en la Ec. (3.12).

$$MTBF_{mp} = \int_0^{\infty} (2e^{-y/200} - e^{-2y/200}) dy = 300 \text{ h} \quad (4.68)$$

■

Modelo II de optimización de inspección

Este modelo tiene por objetivo calcular el número óptimo de inspecciones por instalación de mantenimiento, minimizando el tiempo de ocupación de la instalación por unidad de tiempo. Si TDT es el tiempo destinado a una instalación para hacer un mantenimiento preventivo por unidad de tiempo, entonces [Dhillon (2002)]:

$$TDT(n) = DT_r + DT_i = \frac{\lambda(n)}{\mu} + \frac{n}{\theta} \quad (4.69)$$

Donde:

- TDT(n) - tiempo de parada total del equipo o de la instalación por unidad de tiempo;
- DT_i - tiempo de parada del equipo o de la instalación por unidad de tiempo para inspección;
- DT_r - tiempo de parada del equipo o de la instalación por unidad de tiempo para arreglos;
- n - frecuencia de inspección;
- $\lambda(n)$ - tasa de falla del equipo;
- μ - tasa de reparación del equipo;
- $\frac{1}{\theta}$ - media de los tiempos de inspección exponencialmente distribuidos.

Derivando la Ec. (4.69) respecto n e igualando a cero, se tiene:

$$\frac{d\lambda(n)}{dn} = -\frac{\mu}{\theta} \quad (4.70)$$

■ **Ejemplo 4.9** Considere que la tasa de falla de un sistema es definida por $\lambda(n) = fe^{-n}$. f es la tasa de falla cuando $n=0$. Sustituyendo en la Ec. (4.70), entonces:

$$-fe^{-n} = -\frac{\mu}{\theta} \Rightarrow n^* = \ln \left[\frac{f\theta}{\mu} \right] \quad (4.71)$$

Asumiéndose, por ejemplo, que $\frac{1}{\mu} = 0,02$ mes, $\frac{1}{\theta} = 0,005$ mes y $f=1$ falla por mes, la frecuencia de inspecciones puede ser calculada por la Ec. (4.71).

$$n^* = \ln \left[\frac{1 \cdot 0,02}{0,005} \right] = 1,39 \quad (4.72)$$

Luego, 1 inspección por mes es el número óptimo para inspecciones en el sistema.



Modelo III de optimización de inspección

Este modelo tiene por objetivo calcular el número óptimo de inspecciones por instalación de mantenimiento, considerando la obtención del máximo lucro [Dhillon (2002)]. El modelo considera que un equipo parado no genera lucro. Para ese modelo, serán consideradas las siguientes premisas:

- La tasa de falla del equipo es una función del número de inspecciones;
- Los tiempos para inspección siguen una distribución exponencial;
- Las tasas de falla y de arreglo son constantes.

Además de los términos definidos para la Ec. (4.69), este modelo tiene los siguientes símbolos:

- p - lucro sin pérdidas por parada del equipo;
- C_i - costo medio de inspección por unidad de tiempo sin interrupciones;
- C_r - costo medio de reparos por unidad de tiempo sin interrupciones;
- PL_i - lucro cesante por unidad de tiempo debido a inspecciones;
- PL_r - lucro cesante por unidad de tiempo debido a reparos;
- IC - costo de inspección por unidad de tiempo;
- RC - costo de reparo por unidad de tiempo.

El lucro por unidad de tiempo es calculado por:

$$PR = p - PL_i - PL_r - IC - RC = p - \frac{pn}{\theta} - \frac{p\lambda(n)}{\mu} - \frac{nc_i}{\theta} - \frac{C_r\lambda(n)}{\mu} \quad (4.73)$$

Derivando la Ec. (4.73) con respecto a n e igualando a cero, el lucro tendrá su máximo valor.

$$\frac{d\lambda(n)}{dn} = - \frac{\left[\frac{1}{\theta}(p + C_i) \right]}{\left(\frac{p}{\mu} + \frac{C_r}{\mu} \right)} \quad (4.74)$$

■ **Ejemplo 4.10** Este ejemplo, así como los demás en esta sección, fueron adaptados de [Dhillon (2002)]. Con los datos del ejemplo anterior, se debe calcular el valor óptimo de la frecuencia de inspecciones n .

Sustituyendo la tasa de fallas del sistema del ejemplo anterior, $\lambda(n) = fe^{-n}$, en la Ec. (4.74), se obtiene:

$$n^* = \ln \left[\frac{f\theta(p+C_i)}{\mu(p+C_i)} \right] \quad (4.75)$$

Adoptándose los siguientes valores numéricos, es posible calcular el número óptimo de inspecciones mensuales con base en los costos.

- $p = \$10000$ por mes;
- $f = 2$ fallas por mes;
- $C_i = \$75$ por mes;
- $C_r = \$400$ por mes;
- $\frac{1}{\mu} = 0,04$ mes;
- $\frac{1}{\theta} = 0,01$ mes.

Empleándose la Ec. (4.75), se obtiene:

$$n^* = \ln \left[\frac{2 \cdot 0,04(10000 + 400)}{0,01(10000 + 75)} \right] = 2,11 \quad (4.76)$$

Para el mayor lucro, son necesarias aproximadamente 2 inspecciones por mes.

■

Modelo de Markov para el mantenimiento preventivo

En este modelo, el sistema puede fallar con o sin mantenimiento periódico [Dhillon (2002)]. Si falla, es sometido a un arreglo. De acuerdo con este procedimiento matemático, se puede predecir la disponibilidad del sistema, la probabilidad del sistema parar por mantenimiento preventivo (MP) y su probabilidad de falla.

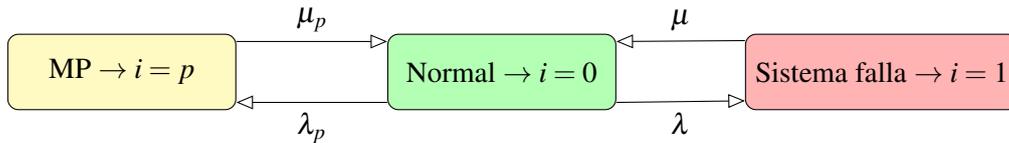


Figura 4.7. Diagrama de transición de sistema

Las siguientes ecuaciones modelan la probabilidad de fallas del sistema.

$$P_0(t) = \frac{\mu_p \mu}{m_1 m_2} + \left[\frac{(m_1 + \mu_p)(m_1 + \mu)}{m_1(m_1 - m_2)} \right] e^{m_1 t} - \left[\frac{(m_2 + \mu_p)(m_2 + \mu)}{m_2(m_1 - m_2)} \right] e^{m_2 t} \quad (4.77)$$

$$P_1(t) = \frac{\mu_p \lambda}{m_1 m_2} + \left[\frac{\lambda m_1 + \lambda \mu_p}{m_1(m_1 - m_2)} \right] e^{m_1 t} - \left[\frac{(m_2 + \mu_p) \lambda}{m_2(m_1 - m_2)} \right] e^{m_2 t} \quad (4.78)$$

$$P_p(t) = \frac{\mu \lambda_p}{m_1 m_2} + \left[\frac{\lambda_p m_1 + \lambda_p \mu}{m_1(m_1 - m_2)} \right] e^{m_1 t} - \left[\frac{(m_2 + \mu) \lambda_p}{m_2(m_1 - m_2)} \right] e^{m_2 t} \quad (4.79)$$

Donde:

$$m_1, m_2 = \frac{-B \pm [B^2 - 4(\mu_p \mu + \lambda \mu_p + \lambda_p \mu)]^{\frac{1}{2}}}{2} \quad (4.80)$$

$$B = \mu + \mu_p + \lambda + \lambda_p \quad (4.81)$$

$$m_1 + m_2 = -B \quad (4.82)$$

$$m_1 m_2 = \mu_p \mu + \lambda_p \mu + \lambda \mu_p \quad (4.83)$$

La disponibilidad del sistema en régimen permanente es dada por la siguiente ecuación:

$$A_{mp} = P_0(t \rightarrow \infty) = \frac{\mu \mu_p}{\mu_p \mu + \lambda_p \mu + \lambda \mu_p} \quad (4.84)$$

■ **Ejemplo 4.11** Considere que un sistema sometido al mantenimiento preventivo, como el presentado en esta sección, tenga los siguientes parámetros: $\lambda = 0,005$ fph, $\lambda_p = 0,008$ por hora, $\mu = 0,009$ reparos por hora y $\mu_p = 0,009$ reparos por hora. Su disponibilidad cuando el tiempo se queda demasiado largo es calculada por la Ec. (4.84).

$$A_{mp} = \frac{0,009 * 0,009}{0,009 * 0,009 + 0,008 * 0,009 + 0,005 * 0,009} = 0,4091 \quad (4.85)$$

■

4.3 Mantenimiento predictivo

El mantenimiento predictivo, citado por [NASA (2000)] como *Predictive Testing & Inspection* (PT&I), es una técnica de monitoreo de condiciones que emplea principalmente test sin intrusión, inspecciones visuales y análisis de desempeño de máquina a partir de sus datos generados. A partir de las informaciones del mantenimiento predictivo, las actividades de mantenimiento preventivo, basadas en calendario, pueden ser modificadas arbitrariamente, adelantando o retrasando un procedimiento dependiendo de las condiciones del sistema.

Uno de los principales retos del mantenimiento predictivo es la colecta de datos que permitan determinar las condiciones del sistema e identificar tendencias de falla. Es una técnica avanzada de mantenimiento y debe ser combinada con otras técnicas.

Sus técnicas más comunes son: mediciones de flujo con equipos de precisión, mediciones de temperatura (incluso la termografía infrarroja), mediciones de presión, mediciones eléctricas, mediciones de espesor de tuberías por ultrasonido, monitoreo de vibraciones en frecuencias específicas, análisis de lubricantes (detección de contaminación y partículas de desgaste), inspecciones por fibra óptica, radiografía, corrientes parásitas, ultrasonido ambiental para la detección de fuga de aire en sistemas de control y en compresores, fotografía y filmación.

Una vez que trata de técnicas específicas, puede ser un mantenimiento costoso, a ser aplicado en sistemas cuyos daños justifiquen los costos de monitoreo.

Según [DECEX (2017)], el conjunto de diagnósticos es basado en parámetros técnicos del equipo, así como parámetros estadísticos de confiabilidad. Midiéndose los parámetros de desempeño, se pueden analizar determinados síntomas de inicio de una falla

Considerando que cada técnica específica tiene sus particularidades delante el sistema que será monitoreado, los diversos métodos de monitoreo de condiciones no serán explorados en este libro. Sin embargo, los tiempos destinados a la inspección pueden ser modelados según las ecuaciones y modelos destinados al mantenimiento preventivo.

Mantenimiento bajo condiciones (*on condition*)

Según [NAVAIR (2005)], el mantenimiento *on condition* es definido como una inspección periódica o continua diseñada para detectar condiciones de falla y permitir una corrección antes de la falla funcional.

Un mantenimiento *on condition* realiza inspecciones y, si ninguna condición potencial de falla se presenta, nada es ejecutado hasta la próxima inspección y el equipamiento permanece en servicio. Si algo es detectado, el mantenimiento *on condition* permite mantener el ítem en servicio por el máximo tiempo posible, minimizando los costos de arreglo y de repuestos necesarios.

Este tipo de mantenimiento es realizado por técnicas que van desde la inspección visual hasta inspecciones complejas no destructivas.

La Fig. 4.8 ilustra la evolución del mantenimiento *on condition*.

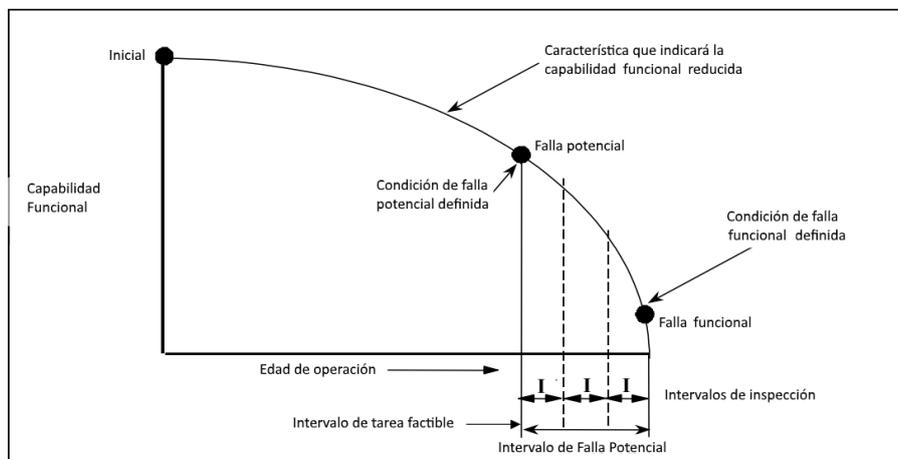


Figura 4.8. Tareas de mantenimiento *on condition*. Copyright 2005 por Direction of Commander, Naval Air Systems Command

El propósito de una tarea *on condition* es definir un nivel aceptable para la probabilidad de una falla potencial, para que su detección ocurra antes de una falla funcional. Por lo tanto, se deben establecer intervalos de inspección (I), un intervalo de falla potencial hasta la falla funcional (FP) y el número de inspecciones (n).

$$I = \frac{FP}{n} \quad (4.86)$$

El cálculo del costo del mantenimiento *on condition* es dado por la Ec. (4.87).

$$OC_{op} = \frac{C_{oc}}{I_{oc}} * \frac{L - (I_i - I_{oc})}{L} + \frac{C_r}{MTBF} \quad (4.87)$$

Donde:

- OC_{op} - costo de la tarea *on condition* por tiempo de operación;
- C_{oc} - costo de una inspección (costo de material, del trabajo, pero no incluye el costo de reparación);
- L - vida de diseño del ítem;
- I_i - intervalo inicial de inspección, que es el intervalo de tiempo hasta la primera inspección;
- I_{oc} - intervalo entre las tareas;

- C_r - costo medio de reparación, que incluye el arreglo de las fallas potenciales y funcionales.

Para fines de cálculo de probabilidad de fallas aceptable (P_{acc}), será adoptada la Tabla 4.1, empleada en un ejemplo perteneciente a [NAVAIR (2005)]. Otros valores podrán ser adoptados, dependiendo del tipo de sistema al que los cálculos serán aplicados.

Tabla 4.1

Probabilidad de fallas aceptable

Clasificación de la severidad	Probabilidad (P_{acc})
I	0,000001
II	0,00001
III	0,0001
IV	0,001

Entonces, para la definición del intervalo de inspecciones *on condition* con base en la consecuencia de las fallas, se debe usar la Ec. (4.86), cuyo valor de n es calculado por la Ec. (4.88), donde θ es la probabilidad de detección de una falla en una inspección.

$$n = \frac{\ln P_{acc}}{\ln (1-\theta)} \quad (4.88)$$

Para el cálculo del intervalo de inspecciones con base en las consecuencias operacionales o económicas, se debe emplear la siguiente ecuación:

$$n = \frac{\ln \left(\frac{-\frac{MTBF * C_i}{FP}}{(C_{npm} - C_{fp}) * \ln (1-\theta)} \right)}{\ln (1-\theta)} \quad (4.89)$$

Donde:

- C_i - costo de la tarea preventiva;
- $C_{npm} = C_{cm} + C_{opc}$ - costo de no hacerse el mantenimiento preventivo;
- C_{cm} - costo de mantenimiento correctivo;
- C_{opc} - costo debido al impacto operacional
- C_{fp} - costo de corrección de falla potencial.

4.4 Mantenimiento modificador

El mantenimiento modificador es definido por [DECEX (2017)] como aquel que consiste en las acciones destinadas a adecuar el equipo a las necesidades dictadas por las exigencias operacionales y mejorar el desempeño de los equipos existentes.

La Fig. 4.9 presenta un esquema de cómo el mantenimiento modificador puede aumentar el desempeño del material por medio del *feedback* del sistema de mantenimiento.

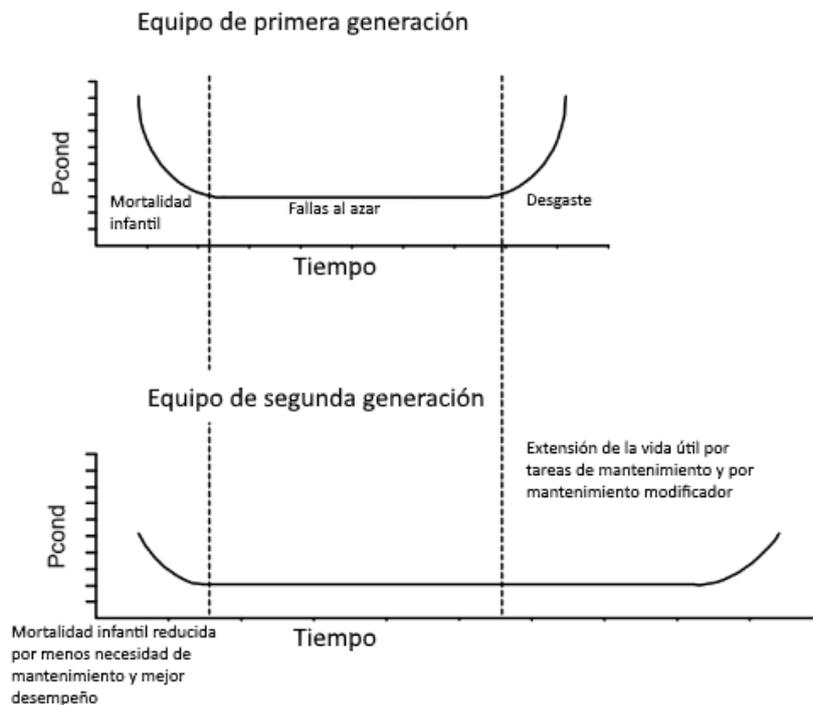


Figura 4.9. Mejoras de diseño por *feedback* de mantenimiento. Copyright 2000 por National Aeronautics and Space Administration

4.4.1 Confiabilidad creciente

La confiabilidad creciente es la mejora en la confiabilidad de un producto (componente, subsistema o sistema) durante un período de tiempo debido a cambios en el diseño del producto y/o el proceso de fabricación [RELIASOFT (2015)].

Según [NAP (2015)], el concepto de confiabilidad creciente se puede interpretar de manera más amplia para abarcar las mejoras de confiabilidad realizadas en un diseño inicial del sistema antes de que se realicen pruebas físicas, es decir, en la fase de diseño, basadas en evaluaciones analíticas. Dicha perspectiva puede ser útil para sistemas que no son susceptibles de pruebas operativas (por ejemplo, satélites).

En el contexto de sistemas ya fabricados y de uso común, la mejora de confiabilidad también es aplicable en aquellos que necesiten modificaciones a lo largo de sus subsecuentes ciclos de vida, con el objetivo de aumentar sus capacidades. Su proceso de mejora está basado en el acrónimo TAAF (del inglés, *test, analyze and fix*). Ese proceso busca identificar potenciales de mejora en la confiabilidad y la consecuente incorporación de actualizaciones específicas en el diseño. Enseguida, es ejecutado un nuevo test para verificar si los modos de falla fueron removidos o mitigados, y para verificar si nuevos modos de falla fueron introducidos.

Los modelos de sistema reparables consideran que la tasa de reparo es constante, sin considerar las mejoras en el sistema ni los cambios de la tasa de reparos (tendencias). Estos modelos ya fueron presentados en la Sección 3.1.1 (ver proceso de Poisson homogéneo).

Cuando se trata de confiabilidad creciente, en la cual el sistema presenta tasas de reparo variables, hay muchos modelos capaces de representar la realidad. Los procesos de Poisson no homogéneos, presentados en la Sección 3.1.1, son adecuados para modelar las tendencias de las tasas de reparo.

De acuerdo con [NAP (2015)], los modelos de confiabilidad creciente generalmente asumen que el único cambio entre sucesivos eventos de pruebas de desarrollo son las mejoras de diseño de confiabilidad del sistema introducidas entre los sucesos. Este supuesto restringe su aplicabilidad porque excluye específicamente la integración de datos de confiabilidad obtenidos de circunstancias de prueba sustancialmente diferentes (dentro de una prueba o en eventos de prueba). Por ejemplo, supongamos que un sistema se prueba primero a bajas temperaturas y que algunos modos de falla se encuentran y se arreglan. Si la próxima prueba es a altas temperaturas, la confiabilidad puede disminuir, aunque el sistema tuvo menos modos de falla debido a las mejoras en el diseño. Debido a que la mayoría de los sistemas están diseñados para una variedad de entornos, se podría argumentar que debería haber curvas de confiabilidad crecientes, separadas y específicas para cada entorno. Esta idea puede ser un tanto extrema, pero es fundamental tener en cuenta que el crecimiento de la confiabilidad es específico de las condiciones de uso.

Si la tasa de fallas sigue un proceso del tipo NHPP(PL), presentado en la Sección 3.1.1, entonces, por medio de la evaluación del parámetro β se puede concluir si la confiabilidad es creciente ($\beta < 1$) o decreciente ($\beta > 1$). La Fig. 3.4 presenta una simulación de un sistema con confiabilidad creciente respecto al tiempo.

[NIST/SEMATECH (2012)] informa que en el año de 1964, J. T. Duane observó que en un sistema sometido a varios test de aumento de confiabilidad, las gráficas de MTBF cumulativas *versus* tiempos de falla en un plan cartesiano del tipo “log-log” presentaban datos que tendían a una línea recta. Esas gráficas, nombradas “gráficas de Duane”, suministran el valor del coeficiente angular ajustado β , que es nombrado “la pendiente de la confiabilidad creciente”.

El MTBF de un sistema sometido a un proceso de test de confiabilidad creciente según un proceso del tipo NHPP(PL), es calculado por la siguiente ecuación:

$$MTBF = \frac{T}{r(1-\beta)} \quad (4.90)$$

Sus intervalos de confianza inferior (M_L) y superior (M_U) son calculados por las siguientes ecuaciones:

$$M_L = MTBF \frac{r(r-1)}{\left[r + \frac{z_{1-\alpha/2}^2}{4} + \sqrt{r \frac{z_{1-\alpha/2}^2}{2} + \frac{z_{1-\alpha/2}^4}{16}} \right]^2} \quad (4.91)$$

$$M_U = MTBF \frac{r(r-1)}{\left[r - z_{1-\alpha/2} \sqrt{\frac{r}{2}} \right]^2} \quad (4.92)$$

En las Ec. (4.91) y (4.92), el $z_{1-\alpha/2}$ es el valor del $100(1 - \alpha/2)$ punto percentil, que puede ser fácilmente obtenido con el método `norm.ppf()` de la biblioteca Scipy Stats.

Como regla práctica, [NIST/SEMATECH (2012)] cita con base en test desarrollados por diversas industrias que el rango $0,3 < \beta < 0,6$ es aplicable para una grande variedad de test. El valor 0,3 se aplica a un test mínimo efectivo, donde tal vez un grupo de estudios multidisciplinario sea inexperto, o en el caso que ni todos los mecanismos de falla sean bien comprendidos. El valor 0,6 indica que los test se aproximan del estado del arte para los desarrollos de confiabilidad creciente.

Sea la ecuación de tasa de reparo o ROCOF descrita en la Sección 3.1.1. La estimación de sus parámetros α y β puede ser obtenida por las siguientes ecuaciones, donde r es el número de fallas ocurridas en el tiempo total T , a t_1, t_2, \dots, t_r [NIST/SEMATECH (2012)];

$$\hat{\alpha} = \frac{r}{T^{1-\beta}} \quad (4.93)$$

$$\hat{\beta} = 1 - \frac{r-1}{\sum_{i=1}^r \ln\left(\frac{T}{t_i}\right)} \quad (4.94)$$

■ **Ejemplo 4.12** [NIST/SEMATECH (2012)] - Sea un sistema sometido a un test de confiabilidad creciente por 1500 horas (aproximadamente 10 semanas), que presentó 8 fallas, según la Tabla 4.2. Se desea calcular el valor de la pendiente β para los datos de ensayos.

Tabla 4.2

Histórico de fallas en un test de confiabilidad creciente

Falla	Horas	MTBF acumulativo
1	33	33
2	76	38
3	145	48,3
4	347	86,8
5	555	111
6	811	135,2
7	1212	173,1
8	1499	187,3

El código 4.3 emplea la biblioteca Numpy, con el método Polyfit, que se basa en el método de los mínimos cuadrados para la obtención de los parámetros de ajuste.

La salida del código fue: “El MTBF estimado al final de las 1500 h de test es de 369 horas”.

El valor calculado por la Ec. (4.94) fue $\hat{\beta} = 0,437$, y con el método numérico Polyfit fue $\beta = 0,493$, en buen acuerdo con el método analítico.

```

# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt

h=1500
tfal=np.array([33,76,145,347,555,811,1212,1499])
r=len(tfal)
fal=np.linspace(1,r,num=r)
cmtbf=np.zeros(r)
cmtbf[0]=tfal[0]
for i in range(1,r):
    cmtbf[i]=((tfal[i]-tfal[i-1])-cmtbf[i-1]+cmtbf[i])/(i+1)+cmtbf[i-1])
beta,b=np.polyfit(np.log(tfal),np.log(cmtbf),1)
x=np.linspace(min(tfal),max(tfal))
y=np.exp(beta*np.log(x)+b)

plt.loglog(x,y,label=r"NHPP en Ley de Potencia, $\beta$=%.3f" %beta)
plt.loglog(tfal,cmtbf,'bo',label="Datos")
plt.xlabel("Edad del sistema")
plt.ylabel("MTBF Acumulativo")
plt.legend(loc='best',frameon=False)
plt.show()

mtbf=h/(r*(1-beta))
print("El MTBF estimado al final de las %i h de teste es de %i horas" %(h,
    mtbf))

```

Código 4.3: Cálculo de tasa de fallas para un sistema con redundancias distintas

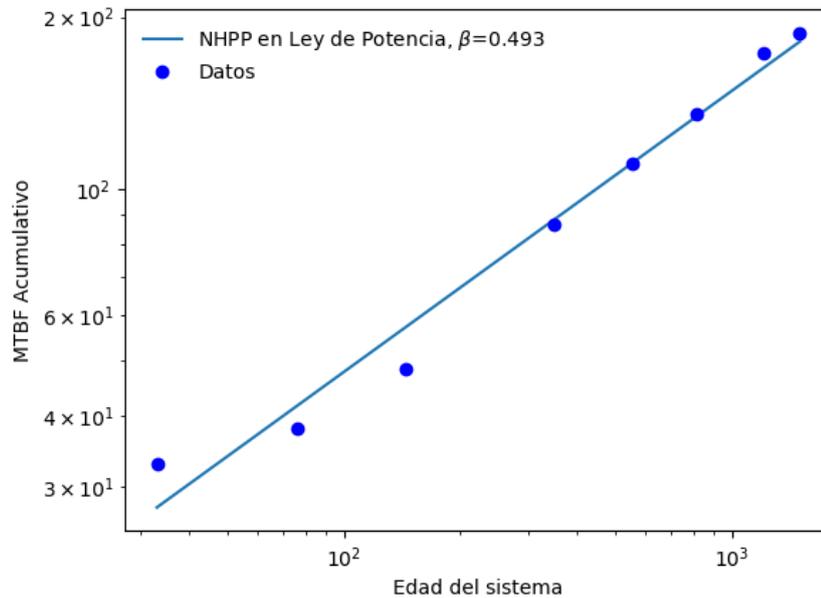


Figura 4.10. Gráfica de Duane – Adaptado de [NIST/SEMATECH (2012)]

■

Otras preguntas podrían ser hechas, variándose el objetivo del ejemplo:

- Dadas 1500 horas de test para alcanzar un MTBF de 370 horas, ¿cuántas fallas se esperan durante los ensayos?
- Considerándose 8 fallas para alcanzar un MTBF de 370 horas, ¿cuántas horas de desarrollo deberán ser dedicadas a los ensayos?

■ **Ejemplo 4.13** A continuación del ejemplo anterior, en este ejemplo será calculado el intervalo de confianza al fin del test para los datos presentados en la Tab. 4.2.

El código 4.4 calcula el ajuste del MTBF y los intervalos de confianza bilaterales para un nivel de 80% y unilateral inferior para un nivel de 90%.

Las salidas fueron:

- El MTBF estimado al final de las 1500 h de test es de 332 horas;
- Los límites para un intervalo de confianza bilateral de 80% son 153.82 y 630.38 horas;
- El límite para un intervalo de confianza unilateral inferior de 90% es 153.82 horas.

```

# -*- coding: utf-8 -*-      1
import numpy as np          2
import scipy.stats as st    3
# %%                          4
h=1500                       5
tfal=np.array([33,76,145,347,555,811,1212,1499]) 6
r=len(tfal)                  7
fal=np.linspace(1,r,num=r)  8
beta=1-(r-1)/(np.sum(np.log(h/ tfal))) 9
                               10
alfa=r/(h**(1-beta))        11
mtbf=h/(r*(1-beta))         12
def ml(mtbf,r,z):           13
    return mtbf*(r*(r-1))/((r+(z**2)/4+np.sqrt(r*((z**2)/2)+((z**4)/16))) 14
    **2)
                               15
def mu(mtbf,r,z):           16
    return mtbf*(r*(r-1))/((r-z*np.sqrt(r/2))**2) 17
                               18
conf=0.90                    19
z90=st.norm.ppf(0.90)       20
                               21
minf=ml(mtbf,r,z90)         22
msup=mu(mtbf,r,z90)         23
                               24
print("El MTBF estimado al final de las %i h de test es de %i horas" %(h, 25
    mtbf))
print("Los limites para un intervalo de confianza bilateral de 80%% son (%.26
    f,%.2f) horas" %(minf,msup))
print("El limite para un intervalo de confianza unilateral inferior de 90%% 27
    es de %.2f horas" %minf)

```

Código 4.4: Ajuste de MTBF e intervalos de confianza



4.4.2 Modelos de aceleración física

La aceleración física permite que un material sea ensayado en condiciones más agresivas, llegando a un estado de fallas anticipadamente. Para eso, son necesarias cargas más grandes para que los síntomas de fallas se presenten más temprano.

Las fallas pueden ocurrir por diversos factores, como una corrosión, la fatiga mecánica, reacciones químicas etc.

Generalmente, las relaciones de aceleración son lineales con el factor de aceleración (AF). Eso quiere decir, por ejemplo, que se un material tiene una cierta probabilidad de falla por fatiga de N ciclos a una tensión de intensidad media T , al multiplicar la intensidad media por 2, se espera que las fallas presenten la misma probabilidad de ocurrencia a $N/2$ ciclos, según una distribución de probabilidad con los mismos valores de parámetros de forma.

Se presentan abajo las principales relaciones de aceleración lineal [NIST/SEMATECH (2012)].

- **Tiempo para fallar** - $t_u = AF * t_s$
- **Probabilidad de falla** - $F_u(t) = F_s(\frac{t}{AF})$
- **Confabilidad** - $R_u(t) = R_s(\frac{t}{AF})$
- **pdf** - $f_u(t) = (\frac{1}{AF})f_s(\frac{t}{AF})$
- **Tasa de falla** - $h_u(t) = (\frac{1}{AF})h_s(\frac{t}{AF})$

Existen diversos modelos de aceleración y [NIST/SEMATECH (2012)] discute respecto algunos de ellos, en especial, involucrando procesos químicos. En esta sección, será presentado de manera muy breve el modelo de crecimiento de grieta mecánica de Coffin-Manson.

El modelo de Coffin-Manson tiene presentado buenos resultados para crecimiento de grieta en soldaduras y en metales sometidos a ciclos de temperatura, como en el caso del ciclo “encender y apagar” [NIST/SEMATECH (2012)].

$$N_f = A f^{-\alpha} \Delta T^{-\beta} G(T_{max}) \quad (4.95)$$

Donde:

- N_f - número de ciclos hasta la falla;
- f - frecuencia del ciclo;
- ΔT - rango de temperatura durante el ciclo;
- $G(T_{max})$ - término de Arrhenius (alrededor de 1,25), que debe ser calculado a la máxima temperatura alcanzada en cada ciclo;
- α - alrededor de -1/3;
- β - alrededor de 2;
- ΔH - energía de activación (puede variar desde 0,3 hasta valores superiores a 1,5).

El modelo de Arrhenius es dado por la Ec. (4.96), donde k es la constante de Boltzmann ($8,617 * 10^{-5}$ eV/K), A es una constante empírica del material y T está en Kelvin.

$$t_f = A e^{\frac{\Delta H}{kT}} \quad (4.96)$$

■ **Ejemplo 4.14** Para calcular un factor de aceleración con base en la Ec. (4.96), es necesario hacerse la relación entre t_1 y t_2 , donde t_i es el tiempo de falla. Entonces:

$$AF = e^{\frac{\Delta H}{k} \left(\frac{1}{T_1} - \frac{1}{T_2} \right)} \quad (4.97)$$

El valor de ΔH depende del mecanismo de falla y de los materiales involucrados.

■



5. Operaciones de Mantenimiento

5.1 Mantenibilidad

[NASA (2000)] define mantenibilidad como la capacidad de conservar o restaurar una función de un sistema dentro de un período de tiempo, cuando es suministrado con un nivel identificado de herramientas, entrenamiento y procedimientos. Se distingue del mantenimiento, que es definido por una acción para conservar una función (es decir, prevenir fallas).

Según [Dhillon (2002)], el término mantenibilidad empezó a ser utilizado en su concepto moderno en la década de 1950 como artículos destinados a diseños de equipos electrónicos, capas, cajas, controles de mantenimiento, instalaciones y acceso al mantenimiento de equipos electrónicos. En 1957, el Departamento de Defensa de los EE.UU. lanzó informes técnicos con recomendaciones que sirvieran de base para normativas referentes a la mantenibilidad. En la década de 1960 surgieron los primeros libros comerciales y normativas militares respecto la mantenibilidad.

5.1.1 Medidas de mantenibilidad

Tiempo medio para reparo (MTTR)

Es una medida del tiempo necesario para proceder a una cierta actividad de mantenimiento. Su cálculo es hecho por la Ec. (4.1).

■ **Ejemplo 5.1** Un sistema es compuesto por 5 subsistemas con las siguientes tasas de falla: $\lambda = (0,0004; 0,0005; 0,0006; 0,0007; 0,0008)$ fph. Los tiempos de mantenimiento correctivo son dados por el siguiente vector: $T = (2; 3; 4; 5; 6)$ horas. El MTTR del sistema es calculado por una media ponderada en razón de las tasas de falla.

$$MTTR = \frac{(0,0004*2)+(0,0005*3)+(0,0006*4)+(0,0007*5)+(0,0008*6)}{0,0004+0,0005+0,0006+0,0007+0,0008} = 4,33 h \quad (5.1)$$

El código 5.1 presentará una rutina interactiva para el cálculo del MTTR.

```

# -*- coding: utf-8 -*-      1
import numpy as np          2
# %%                          3
n=int(input("Entre con la cantidad de subsistemas \n"))      4
lamb=np.zeros(n)           5
t=np.zeros(n)              6
for i in range (n):        7
    lamb[i]=input("Entre con el valor de la tasa de fallas para el  8
    subsistema %d\n" %(i+1))
    t[i]=input("Entre con el valor del tiempo de mantenimiento correctivo  9
    para el subsistema %d\n" %(i+1))
mttr=np.sum(np.multiply(lamb,t))/np.sum(lamb)                10
print("MTTR=%.3f" %mttr)                                     11

```

Código 5.1: Cálculo del MTTR para un sistema con n subsistemas

Tiempo medio de mantenimiento

Está compuesto por las parcelas de tiempo medio de retraso administrativo (MADT), tiempo medio de retraso logístico (MLDT) y por el tiempo necesario para hacerse mantenimientos correctivo y preventivo (Ec. (4.58)).

$$MMD = \bar{M} + MADT + MLDT \quad (5.2)$$

Las funciones de mantenibilidad pueden ser cualquiera de las funciones probabilísticas ya estudiadas, como la exponencial, la de Weibull, la normal, o incluso otras no presentadas en este libro como la gamma y la de Erlang, por ejemplo.

5.1.2 Factores humanos

Los factores humanos tienen gran influencia en la mantenibilidad, aumentando considerablemente los costos del mantenimiento y su tiempo de ejecución.

Datos disponibles en [Dhillon (2002)] informan que:

- 30% de las fallas en equipos electrónicos se deben a errores de operación y de mantenimiento, siendo 12% debidos a condición accidental o anormal, 10% debidos a manoseo y 8% a mantenimiento defectuoso;
- Un estudio de 1993 reveló que en 122 ocurrencias en mantenimiento se involucraron factores humanos, siendo 30% debido a instalaciones incorrectas, 56% referentes a omisiones, 8% debido a componentes incorrectos y otras causas fueron responsables por 6% de los daños;
- Entre 1982 y 1991, el mantenimiento y la inspección fue el segundo problema más importante de seguridad en vuelos internacionales de avión, responsables por 1481 fatalidades;
- Un retraso de una hora de vuelo implica en un perjuicio estimado de US\$ 10000;
- 40 a 50% de los elementos removidos para arreglos en operaciones aéreas comerciales no presentan defectos;
- En 1979 un accidente con un avión modelo DC-10, debido a errores de procedimientos en mantenimiento, mató 272 personas;
- Un incidente en un montaje de válvulas en el campo petrolero de Ekofisk costó US\$ 50 millones;

- Un estudio en errores de montaje en operaciones con misiles reveló problemas con tuercas o conexiones sueltas (14%), instalación incorrecta (28%), cuadrantes y controles (lectura incorrecta, mal ajuste) (38%), inaccesibilidad (3%) y otros errores (17%);
- En 1983, una aeronave modelo L-1011 perdió presión de aceite en tres motores debido a anillos de sellados, debido a inspección mal hecha y procedimientos de suministros;
- Un estudio involucrando varias tareas como ajuste, alineamiento y remoción de componentes indicó que la media de la confiabilidad humana es de 0,9871. En ese sentido, se esperan errores por el personal de mantenimiento en 13 de cada 1000 operaciones;
- Un estudio clasificó el tiempo de mantenimiento en equipos electrónicos en tres grupos y sus porcentuales de fallas: diagnóstico (65-75%), acciones correctivas (15-25%) y verificación (5-15%).

La Fig. 5.1 presentará la evolución de las fallas por errores humanos a lo largo del ciclo de vida de un sistema. Se observa que ocurre un aumento considerable de errores cuando el sistema empieza a envejecer.

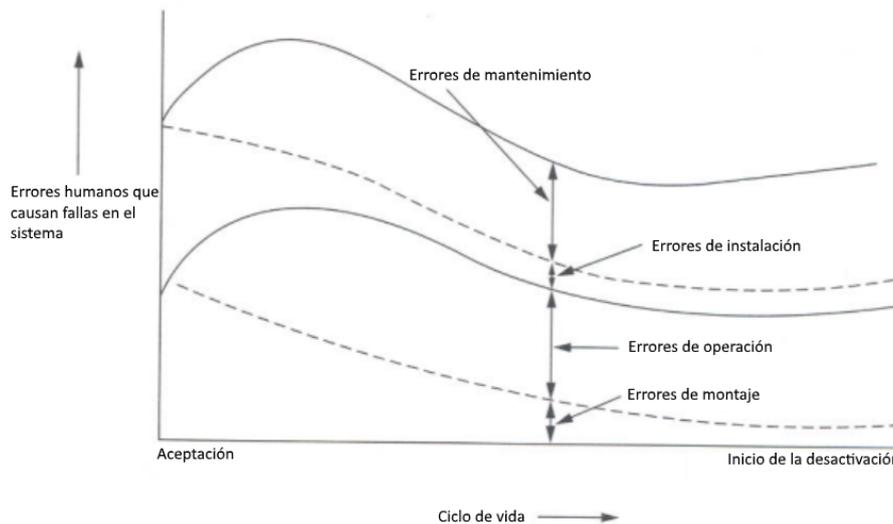


Figura 5.1. Ciclo de vida versus errores humanos. Copyright 2002 por CRC Press

Para cuantificar los errores humanos en términos de confiabilidad, serán presentados 3 modelos matemáticos. Los dos primeros serán basados en cadena de Markov. El último será modelado como un árbol de fallas.

Modelo I

En este modelo [Dhillon (2002)], el sistema puede fallar por errores de mantenimiento (estado 1) o por otros motivos (estado 2). Sus ecuaciones de probabilidades de falla son las mismas que las presentadas en la Sección 4.2.1, asegurados los cambios de los términos.

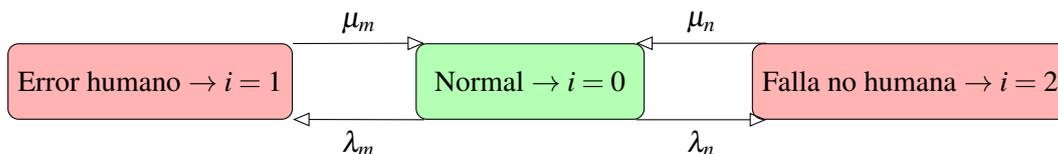


Figura 5.2. Diagrama de transición de sistema

■ **Ejemplo 5.2** Un sistema puede fallar por errores humanos a una tasa de $\lambda_n = 0,0002$ fph o por fallas del equipo a una tasa de $\lambda = 0,005$ fph. Siendo ese sistema modelado por el método de Markov, se desea calcular la probabilidad del sistema fallido por errores humanos en 100 horas de misión.

Siendo $\mu_n = 0$ en la Ec. (4.78), o sea, sin mantenimiento preventivo, entonces:

$$P_1(100) = \frac{0,0002}{0,0002+0,005} (1 - e^{-(0,0002+0,005)*100}) = 0,0156 \quad (5.3)$$

■ **Modelo II**

En este modelo [Dhillon (2002)], el sistema solamente puede fallar debido a problemas del equipamiento, pero su desempeño disminuye por errores de mantenimiento.

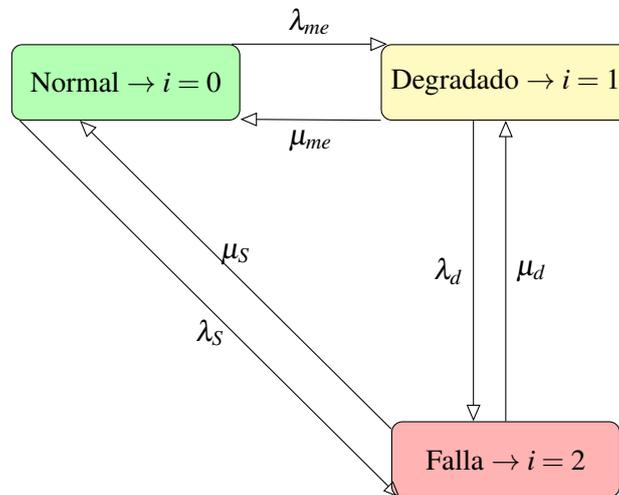


Figura 5.3. Diagrama de transición de sistema

Las siguientes ecuaciones modelan la probabilidad de fallas del sistema.

$$P_0(t) = P1 + P2e^{K_1t} + (1 - P1 - P2)e^{K_2t}$$

$$P1 = \frac{\mu_{me}\mu_s + \lambda_d\mu_s + \mu_{me}\mu_d}{K_1K_2}$$

$$P2 = \left[\frac{\mu_{me}K_1 + \mu_sK_1 + \mu_dK_1 + K_1\lambda_d + K_1^2 + \mu_{me}\mu_s + \lambda_d\mu_s + \mu_{me}\mu_d}{K_1(K_1 - K_2)} \right] \quad (5.4)$$

$$P_1(t) = P3 + P4e^{K_1t} - (P3 + P4)e^{K_2t}$$

$$P3 = \frac{\lambda_{me}\mu_s + \lambda_{me}\mu_d + \lambda_s\mu_d}{K_1K_2}$$

$$P4 = \left[\frac{\lambda_{me}K_1 + \mu_s\lambda_{me} + \mu_d\lambda_{me} + \lambda_s\mu_d}{K_1(K_1 - K_2)} \right] \quad (5.5)$$

$$P_2(t) = P5 + P6e^{K_1t} - (P5 + P6)e^{K_2t}$$

$$P5 = \frac{\lambda_{me}\lambda_d + \lambda_s\mu_{me} + \lambda_s\lambda_d}{K_1K_2}$$

$$P6 = \left[\frac{\lambda_sK_1 + \lambda_d\lambda_{me} + \mu_{me}\lambda_s + \lambda_s\lambda_d}{K_1(K_1 - K_2)} \right] \quad (5.6)$$

Donde:

$$K_1 K_2 = \mu_{me} \mu_s + \lambda_d \mu_s + \mu_{me} \mu_d + \lambda_{me} \mu_s + \lambda_{me} \mu_d + \lambda_{me} \lambda_d + \lambda_s \mu_{me} + \lambda_s \mu_d + \lambda_s \lambda_d \quad (5.7)$$

$$K_1, K_2 = \frac{-B \pm (B^2 - 4K_1 K_2)^{1/2}}{2} \quad (5.8)$$

$$B = \mu_{me} + \mu_s + \mu_d + \lambda_{me} + \lambda_s + \lambda_d \quad (5.9)$$

La probabilidad de fallo del sistema por degradación debido a errores humanos, a un tiempo muy largo, se queda: $P_1(t \rightarrow \infty) = P_3$

Para evaluar la disponibilidad del sistema en operación, la disponibilidad se queda: $A_s(t) = P_0(t) + P_1(t)$. Así, cuando $t \rightarrow \infty$, las disponibilidad se queda:

$$A_s = \frac{\mu_{me} \mu_s + \lambda_d \mu_s + \mu_{me} \mu_d + \lambda_{me} \mu_s + \lambda_{me} \mu_d + \lambda_s \mu_d}{K_1 K_2} \quad (5.10)$$

■ **Ejemplo 5.3** El Código 5.2 presentará el cálculo de la probabilidad de un sistema sometido a degradación por errores humanos, en estado permanente. Para ese sistema, son dados los siguientes parámetros: $\lambda_s = 0,004$ fallas/h, $\lambda_{me} = 0,0001$ fallas/h, $\lambda_d = 0,001$ fallas/h, $\mu_s = 0,02$ arreglos/h, $\mu_{me} = 0,008$ y $\mu_d = 0,06$ arreglos/h.

```
# -*- coding: utf-8 -*-
import numpy as np
lamb=np. array ([0.004 ,0.0001 ,0.001])
mu=np. array ([0.02 ,0.008 ,0.06])
k1k2=mu[0]*mu[1]+mu[0]*lamb[2]+mu[1]*mu[2]+mu[0]*lamb[1]+lamb[1]*mu[2]+lamb
[1]*lamb[2]+lamb[0]*mu[1]+lamb[0]*mu[2]+lamb[0]*lamb[2]
p1ss=(lamb[1]*mu[0]+lamb[1]*mu[2]+lamb[0]*mu[2])/k1k2
print('La probabilidad del sistema degradarse debido a errores de
mantenimiento es %.4f' %p1ss)
```

Código 5.2: Cálculo de disponibilidad para un sistema con posibilidad de operación parcial

La salida del código fue: “La probabilidad del sistema degradarse debido a errores de mantenimiento es 0,2627”.

■

Método de análisis del árbol de falla

El método de análisis del árbol de falla (del inglés, *fault tree analysis* - FTA) fue desarrollado en la década de 1960 por los Laboratorios Bell y es una técnica analítica, por la cual se especifica un estado no deseado del sistema (generalmente un estado que es crítico desde un punto de vista de seguridad o confiabilidad), y luego el sistema se analiza en el contexto de su entorno y operación para encontrar formas realistas en las que el evento no deseado (evento superior) puede ocurrir [NASA (2002)].

Las fallas pueden ser eventos asociados con fallas de *hardware*, de componentes, errores humanos o cualquier otro evento pertinente que pueda conducir a eventos no deseados [Vesely (1981)].

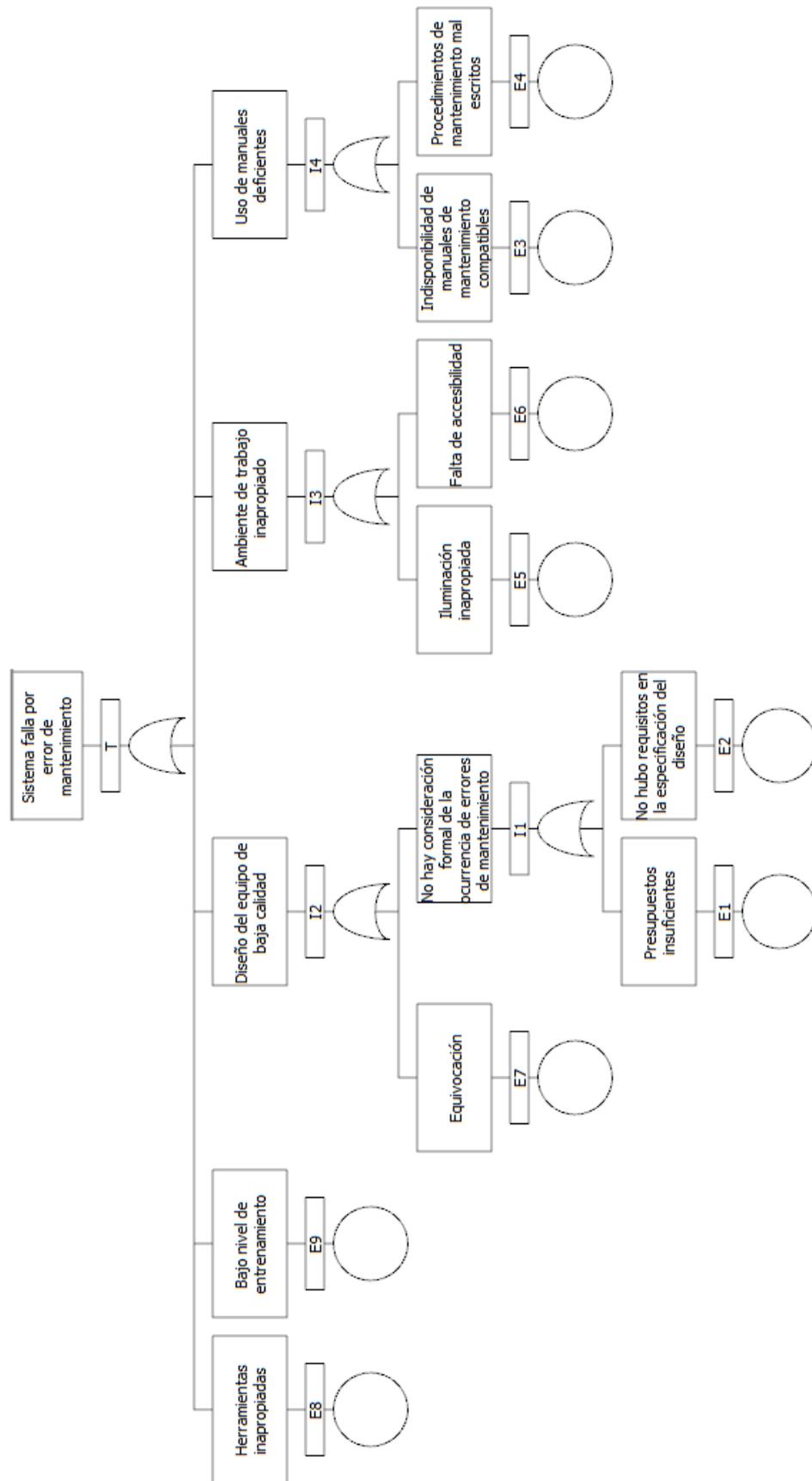


Figura 5.4. Árbol de fallas. Copyright 2002 por CRC Press

■ **Ejemplo 5.4** La Fig. 5.4 presenta un ejemplo de posibles eventos que pueden ocasionar fallas humanas. Para reproducir ese árbol de fallas, fue empleado el *software* libre SCRAM, versión 0.16.2. Ese *software* permite el cálculo de la probabilidad de un árbol, para el evento principal, nombrado *top event* (T).

Para fines de ejemplo, todas las probabilidades de ocurrencias (E_1, E_2, \dots, E_9) fueron asumidas como 0,02. El cálculo del *top event* demandará la obtención de las probabilidades de los eventos I_1 a I_4 .

Como los eventos son conectados por puertas lógicas del tipo *OR* y son independientes, el cálculo será:

$$P(I_1) = P(E_1) + P(E_2) - P(E_1)P(E_2) = 0,0396 \quad (5.11)$$

Lo mismo ocurre para los demás eventos, resultando: $P(I_2) = 0,0588$, $P(I_3) = P(I_4) = P(I_1)$. El *top event* se queda:

$$P(T) = 1 - [1 - P(E_8)][1 - P(E_9)][1 - P(I_2)][(1 - P(E_3))[1 - P(I_4)] = 0,1663 \quad (5.12)$$

La salida del código generado por el programa SCRAM para la probabilidad del sistema fallar por errores de mantenimiento fue la misma que la calculada analíticamente, de aproximadamente 17%.

■

5.2 Control de stock

El control de stock, asociado a los costos de su adquisición, constituye un aspecto importante del proceso productivo de una organización, porque uno de los principales retos del mantenimiento es ahorrar costos y garantizar la eficiencia. A medida que los equipos quedan más complejos, sus repuestos aumentan en términos de costos y de cantidad.

[Dhillon (2002)] informa que el principio de Pareto (homenaje al economista y sociólogo Vilfredo Pareto (1848-1923)) es aplicable al control de repuestos. Según ese principio, los repuestos siguen una clasificación nombrada ABC, en la cual:

- A: donde 20% de los artículos son responsables por 80% de los costos;
- B: donde 30% de los artículos son responsables por 15% de los costos; y
- C: donde 50% de los artículos son responsables por 5% de los costos.

Los costos de mantenerse un stock son presentados en rangos aproximados por la Tabla 5.1. Los porcentajes se refieren al valor del stock, como referencia. Sin embargo, dependiendo del tipo de material que se encuentra en el stock y de su respectivo valor agregado, los porcentajes pueden cambiar.

Tabla 5.1

Costos de mantenimiento de stock

Costo	Actividades	Porcentual del valor del stock
Costo de mantenimiento de stock	Costo de operación, alquiler de edificio, seguros, depreciación, etc.	3-10
Costo de inversión	Costo del préstamo, costo del seguro de inventario, impuestos, etc.	6-24
Costo laboral (asociado con manejo extra)	—	3-5
Manejo de materiales	Arrendamiento de equipos, energía, costos de operación, etc.	1-3,5
Costos diversos	Costes asociados a chatarra, hurto, obsolescencia, etc.	2-5

Además de los costos de mantenimiento de stock, se deben considerar:

- Costo de pedido: está asociado con el procesamiento de pedidos, el soporte administrativo, los formularios, los suministros, etc.
- Costo de instalación (SUC): está asociado con la preparación de un equipo / máquina o proceso para la fabricación de un pedido.

En las próximas secciones serán presentados modelos matemáticos de control de costos en mantenimiento.

5.2.1 Modelo de cantidad económica de compra

En este modelo de control de stocks [Dhillon (2002)], serán asumidas las siguientes condiciones:

- Hay una demanda constante y conocida para los repuestos;
- Los repuestos son abastecidos de inmediato, de una sola vez siempre que el stock se queda vacío;
- Tanto la compra como el recibimiento del repuesto son regularmente distribuidos en el tiempo;
- No hay descuentos;
- Los pedidos de repuestos garantizan que no hayan desabastecimientos; y
- Dos parámetros son considerados cada vez: el costo de se mantener los repuestos a lo largo del tiempo; y el costo de procesamiento del pedido o el costo de la preparación de un equipo o proceso para atender a un pedido.

El costo anual para se preparar el sistema o proceso para atender a un pedido se queda:

$$SUC = NOP * SOC = \frac{\theta}{q} * SOC \quad (5.13)$$

Donde:

- NOP - número de pedidos por año;
- SOC - costo del proceso para atender a un pedido o costo para el procesamiento de un pedido;
- θ - demanda en unidades de un ítem por año;
- q - número de piezas por pedido.

El costo anual para mantener el stock se queda:

$$AHC = AIL * HC = \frac{q}{2} * HC \quad (5.14)$$

Donde:

- AIL - nivel medio del stock;
- HC - costo anual para mantener una unidad.

La Fig. 5.5 ilustrará el comportamiento del nivel de stock.

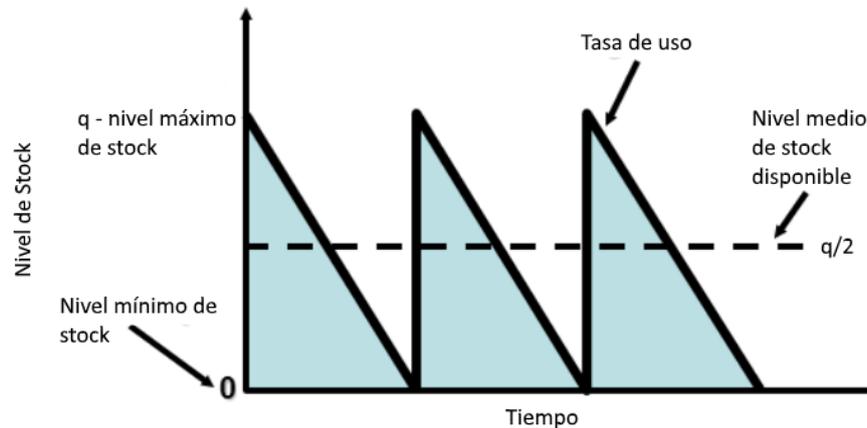


Figura 5.5. Niveles de stock. Copyright 2017 por Canadian Center of Science and Education

Igualando SUC a AHC, se obtiene la cantidad óptima de pedidos:

$$q^* = \sqrt{\frac{2\theta SOC}{HC}} \quad (5.15)$$

En el caso de la Ec. (5.15), q^* es el número óptimo de piezas por pedido o la cantidad óptima de pedidos (EOQ). La Fig. 5.6 ilustrará el comportamiento del costo ideal de stock.

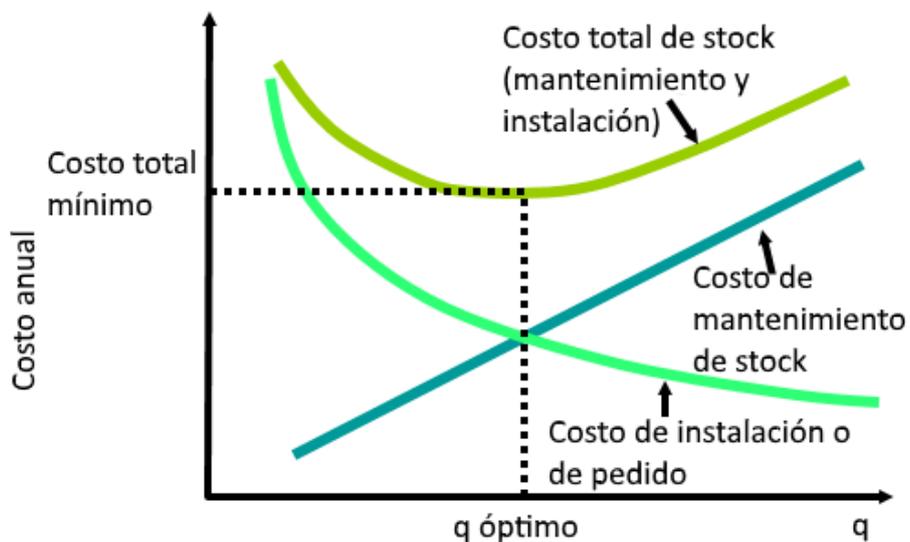


Figura 5.6. Costo ideal de stock. Copyright 2017 por Canadian Center of Science and Education

El número anual esperado de pedidos es dado por:

$$n = \frac{\theta}{q^*} \quad (5.16)$$

El tiempo esperado entre pedidos (ETBO), siendo TWD el número de días laborales por año, es:

$$ETBO = \frac{TWD}{n} \quad (5.17)$$

La demanda diaria (DD) es:

$$DD = \frac{\theta}{TWD} \quad (5.18)$$

Considerándose que hay un retraso (LT) de algunos días para que un pedido sea atendido, y que la demanda es uniforme y constante, el punto para hacerse un nuevo pedido es:

$$ROP = DD * LT \quad (5.19)$$

■ **Ejemplo 5.5** Sea un departamento de mantenimiento que estima el consumo anual de 600 unidades de un ítem. El costo anual para mantener una unidad de este ítem es \$ 1,20 y cada pedido cuesta \$ 5,00.

El número óptimo de unidades por pedido es:

$$q^* = \sqrt{\frac{2*600*5}{1,20}} \approx 71 \quad (5.20)$$

Sabiéndose el número óptimo de piezas por pedido, se puede calcular la cantidad de pedidos por año ($n = 600/71 \approx 8$) y considerándose que en el año hay 250 días laborales, es posible calcular el tiempo medio entre los pedidos, en días ($ETBO = 250/8 \approx 31$).

Considerándose que la entrega de cada uno de esos pedidos lleva 5 días laborales, el momento ideal para hacerse un nuevo pedido es: $DD = 600/250 \approx 3 \Rightarrow ROP = 3 * 5 = 15$. O sea, cuando el stock quedarse con 15 unidades, un nuevo pedido debe ser hecho.

■

5.2.2 Modelo de cantidad de pedidos para la producción

En el modelo anterior, fue asumido que los pedidos son entregados de una sola vez. Sin embargo, debido a cuestiones relacionadas a la capacidad del fabricante producir los pedidos, el tiempo para la entrega implica que el stock del solicitante continua bajando sus niveles hasta la entrega de los pedidos.

Considerándose el equilibrio entre la tasa de consumo de componentes y la tasa de entrega de repuestos, se puede llegar a la siguiente ecuación, donde RT es el tiempo para rellenar el stock y r es la tasa de rellenado (en unidades por día).

$$RT = \frac{q}{r} \quad (5.21)$$

El uso (U) durante el periodo de relleno es calculado por:

$$U = \frac{q}{r}DD \quad (5.22)$$

El nivel máximo de stock (MIL) es calculado por:

$$MIL = q - U = q - \frac{q}{r}DD = q \left(1 - \frac{DD}{r}\right) \quad (5.23)$$

En consecuencia, el costo para mantener el stock se queda:

$$AHC = \frac{MIL}{2}HC = \frac{q}{2} \left(1 - \frac{DD}{r}\right) HC \quad (5.24)$$

Haciéndose iguales las Ec. (5.13) y (5.24), se puede obtener la cantidad óptima de pedidos.

$$q^* = \sqrt{\frac{2\theta SOC}{\left(1 - \frac{DD}{r}\right)HC}} \quad (5.25)$$

■ **Ejemplo 5.6** Sea un sector de mantenimiento que calcula que en el próximo año tendrá una demanda de 1200 componentes. La demanda diaria estimada del sector de mantenimiento será de 10 componentes y la capacidad de entrega del fabricante será de 12 componentes por día. Si el costo de mantenerse cada pieza sea de \$ 1,50 y de hacerse el pedido sea de \$ 6,00, el número óptimo de unidades por pedido es:

$$q^* = \sqrt{\frac{2 * 1200 * 6}{\left(1 - \frac{10}{12}\right)1,50}} = 240 \quad (5.26)$$

■

5.2.3 Modelo de pedidos con descuento

En casos donde las compras de repuestos se hicieran en gran cantidad, es posible que el fabricante permita descuentos. Sin embargo, aunque los costos de adquisición reduzcan, los costos de mantener el stock aumentan. Para eso, se debe calcular el equilibrio entre lo que se gana con el descuento y lo que se gasta con los costos de mantenimiento de stock.

Considerando que hay un costo anual del stock ($TAIC$), un costo de producción (PC), un costo unitario (C_u) y un costo unitario de mantenimiento de los repuestos en stock (iC_u , donde i es una porcentaje del precio (C_u)), las siguientes ecuaciones permiten calcular el número óptimo por pedido [Dhillon (2002)].

$$TAIC = SUC + AHC + PC = \frac{\theta}{q}SOC + \frac{q}{2}HC + \theta C_u \quad (5.27)$$

$$q^* = \sqrt{\frac{2\theta SOC}{iC_u}} \quad (5.28)$$

■ **Ejemplo 5.7** Sea un sector de mantenimiento que estima que en el próximo año tendrá una demanda de 4000 componentes. Hay un costo fijo por cada pedido de \$ 40,00. El costo de mantenimiento en stock para cada componente es de 25 % del valor del componente.

El fabricante permite descuentos en las compras de grandes cantidades, con los siguientes precios:

- De 0 hasta 499 componentes: \$ 10,00 por componente;
- De 500 hasta 999 componentes: \$ 8,00 por componente;
- Arriba de 1000 componentes: \$ 7,00 por componente.

Con el fin de obtenerse el equilibrio de costos, serán calculadas las cantidades óptimas (unidades por pedido), considerando los 3 precios unitarios.

$$q_1^* = \sqrt{\frac{2 \cdot 4000 \cdot 40}{0,25 \cdot 10}} \approx 358 \quad (5.29)$$

$$q_2^* = \sqrt{\frac{2 \cdot 4000 \cdot 40}{0,25 \cdot 8}} \approx 400 \quad (5.30)$$

$$q_3^* = \sqrt{\frac{2 \cdot 4000 \cdot 40}{0,25 \cdot 7}} \approx 428 \quad (5.31)$$

Aplicando los valores para el cálculo del mantenimiento anual del stock en la Ec. (5.27), se calculan los costos de mantenimiento de las siguientes cantidades relativas a los pedidos:

- Pedido de 358 componentes: costo total de las piezas=\$ 40000,00; AHC=\$ 447,50; costo anual de adquisición, por pedido \approx \$ 447,00; costo total \approx \$ 40894,00;
- Pedido de 500 componentes: costo total de las piezas=\$ 32000,00; AHC=\$ 500,00; costo anual de adquisición, por pedido=\$ 320,00; costo total=\$ 32820,00;
- Pedido de 1000 componentes: costo total de las piezas=\$ 28000,00; AHC=\$ 875,00; costo anual de adquisición, por pedido=\$ 160,00; costo total=\$ 29035,00.

La compra en pedidos de 1000 componentes minimiza los gastos.

■

5.2.4 Stock de seguridad

Un stock de seguridad puede ser calculado como una función de las demandas de mantenimiento. Considerando μ como la demanda media durante el tiempo de retraso del pedido y σ su correspondiente desviación estándar, el punto de pedido (ODP) puede ser calculado por:

$$ODP = \mu + z\sigma \quad (5.32)$$

■ **Ejemplo 5.8** Sea un sector de mantenimiento que estima una demanda de un determinado ítem, durante un proceso de adquisición de repuestos, que sigue una distribución normal con media 50 y desviación estándar igual a 4. El stock de seguridad durante un período de pedido, con 95 % de nivel de servicio, debe ser calculado.

Para el nivel de 95 % en una distribución normal, $z=1,64$. Aplicando los valores a la ecuación del stock de seguridad, se obtiene $ODP = 50 + 1,64 \cdot 4 \approx 57$. Es decir, el stock de seguridad debe ser de 7 componentes de repuesto y un nuevo pedido debe ser hecho cuando el nivel de stock llegar a 57 componentes.

■

5.2.5 Disponibilidad de piezas en stock

Uno de los cálculos en mantenimiento que son importantes para su planificación se dedica a la estimación de una cantidad de piezas de repuestos, garantizando la disponibilidad del equipo durante un cierto tiempo. Para ese cálculo, se considera la confiabilidad del ítem en análisis.

Según [Dhillon (2002)], la probabilidad de que un ítem esté disponible cuando sea requerido es bien modelado por una distribución de Poisson.

$$P_s = \sum_{j=0}^n \frac{(-\ln e^{-q\lambda t})^j e^{-q\lambda t}}{j!} \quad (5.33)$$

En la Ec. (5.33), se pueden identificar los siguientes términos:

- P_s - probabilidad de un repuesto estar disponible a un determinado tiempo;
- λ - tasa de falla de la pieza;
- t - tiempo;
- n - número de repuestos en stock;
- q - número de piezas de un tipo específico empleadas.

■ **Ejemplo 5.9** Un equipamiento posee 30 piezas de un determinado tipo, la cual presenta una tasa de falla de 20 fallas por millón de horas de operación. Considere que el equipamiento opera continuamente y las piezas son repuestas a cada 4 meses. Si en el stock hay solamente 4 piezas disponibles, se debe calcular la probabilidad de piezas disponibles en este período.

Considerando el período de 4 meses de servicio continuo, el tiempo puede ser calculado como $t = 4 * 30 * 24 = 2880$ horas.

De la Ec. (5.33), se calcula:

$$P_s = \sum_{j=0}^4 \frac{(-\ln e^{-30*20*10^{-6}*2880})^j e^{-30*20*10^{-6}*2880}}{j!} = 0,9684 \quad (5.34)$$

El código 5.3 calculará la disponibilidad de este ejemplo.

```
# -*- coding: utf-8 -*-
import numpy as np
# %%
n=4
q=30
lamb=20*10**(-6)
t=4*30*24
ps=np.zeros(n+1)
for i in range(n+1):
    ps[i]=((( - np . log ( np . exp(-q*lamb*t) ) ) ** i ) * np . exp(-q*lamb*t) ) / np . math .
    factorial(i)
pss=np.sum(ps)
print("La disponibilidad de piezas en stock sera de %.1f%%%(pss*100))
```

Código 5.3: Cálculo de disponibilidad de pieza de repuesto en stock

Habrá una probabilidad de aproximadamente 97% de disponibilidad de la pieza.



5.3 Costos de mantenimiento

Para una buena planificación del mantenimiento, es importante considerar los presupuestos durante el periodo considerado.

Las siguientes ecuaciones permitirán estimar los costos de mantenimiento en un determinado periodo, considerando una tasa constante de corrección del valor a lo largo del periodo.

Modelo I de valor presente

Esta es la ecuación del interés compuesto. El modelo permite calcular un valor futuro (AM) desde un valor presente (PV) que sufre correcciones periódicas (i) a cada k periodos.

$$PV = \frac{AM}{(1+i)^k} \quad (5.35)$$

Modelo II de valor presente

Con este modelo [Dhillon (2002)] es posible estimar los costos de mantenimiento (MC) al final de k periodos, corregidos por un factor i constante por periodo.

$$PV = MC \frac{1-(1+i)^{-k}}{i} \quad (5.36)$$

■ **Ejemplo 5.10** Un departamento necesita establecer un sistema de mantenimiento y solicita a dos fabricantes de un determinado equipamiento informaciones sobre sus tasas de fallas y sobre el costo del mantenimiento correctivo de esos equipos. Se desea calcular cuál equipamiento es más ventajoso para el departamento. La vida útil estimada para el equipamiento en la institución compradora es de 12 años. Ambos los fabricantes estimarán sus correcciones de costos de mantenimiento correctivo anual en 8 %

El equipamiento del fabricante A tiene un costo actual medio de mantenimiento correctivo de \$ 1000,00 por operación de mantenimiento. El fabricante B tiene un costo establecido en \$ 1400,00.

La tasa de fallas estimada del equipamiento del fabricante A es de 2,5 fallas por año. El equipamiento del fabricante B tiene una tasa de fallas estimada en 2 fallas por año.

Entonces, el costo de mantenimiento anual esperado para el equipamiento del fabricante A es de $AC_{cm} = 1000 * 2,5 = 2500,00$. Para el fabricante B es de $AC_{cm} = 1400 * 2 = 2800,00$

El valor actual, entonces, es calculado como:

$$PV_A = 2500 \frac{1-(1+0,08)^{-12}}{0,08} = 18840,20 \quad (5.37)$$

$$PV_B = 2800 \frac{1-(1+0,08)^{-12}}{0,08} = 21101,01 \quad (5.38)$$

Si se adquiere el equipamiento del fabricante A es posible ahorrar \$ 2260,81 en 12 años, comparándose con el equipamiento del fabricante B.

■

5.3.1 Relación costo beneficio

Según [DMAvEx (2010)], la relación costo beneficio (Rcb) es una variable aleatoria definida como el cociente entre la variable costo (C) y un indicador de desempeño, sea el operacional, de eficacia logística o de confiabilidad.

En las Ec. (5.39) y (5.40), será empleada la confiabilidad (R) del sistema que representa el material.

$$Rcb(T) = \frac{C(T)}{R(T)} \tag{5.39}$$

La tendencia de la Rcb está dada por la derivada de la Rcb con respecto al tiempo. Aplicándose la derivada con respecto al tiempo sobre la Ec. (2.9), se obtiene $\frac{dR}{dt} = -f(t)$, donde f(t) es la pdf, presentada en la Ec. (2.3). Entonces:

$$\frac{dRcb}{dt} = \frac{dR}{dt} \left[\frac{1}{R} \frac{\partial C}{\partial R} - \frac{C}{R^2} \right] = -f(t) \left[\frac{S_{cr}}{R} - \frac{C}{R^2} \right] \tag{5.40}$$

En la Ec. (5.40), $S_{cr} \forall S_{cr} < 0$ es el parámetro de sensibilidad del costo respecto el nivel de confiabilidad. Sin embargo, [DMAvEx (2010)] destaca que las variables C y R no son necesariamente independientes, una vez que:

- para un sistema de alta confiabilidad, los costos directos e indirectos de mantenimiento y de operación del sistema (que son incorporados en la variable C) resultarán en un costo total compensador. Por ejemplo, una aeronave recientemente sometida a un *overhaul* presentará el parámetro S_{cr} atenuado;
- para un sistema de baja confiabilidad, los mismos costos citados en el ítem anterior resultarán en un costo total mayor. Por ejemplo, una aeronave en un fin de ciclo presentará el parámetro S_{cr} más acentuado.

La Fig. 5.7 ilustrará el comportamiento de S_{cr} .

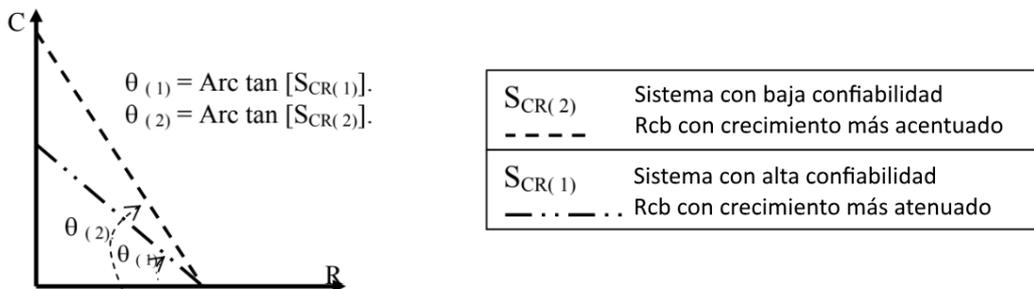


Figura 5.7. Comportamiento del parámetro de sensibilidad S_{cr} . Copyright 2010 por Diretoria de Material de Aviação do Exército

[NAVAIR (2005)] presenta la Ec. (5.41), que establece el costo beneficio para actividades de mantenimiento basadas en tareas operacionales y económicas, desde que esas tareas cuesten menos que las consecuencias para las cuales fueron diseñadas para evitar.

$$Rcb = \frac{C_{bf}N_s + C_{af}(1-N_s)}{N_s t + (1-N_s)MTTF_p} \cdot \frac{C_{af}}{MTBF} \tag{5.41}$$

Donde:

- C_{bf} - costo de rehacer un trabajo o de reponer un componente antes de la falla;
- N_s - porcentaje de piezas que sobrevivirán al intervalo propuesto para la tarea;
- t - intervalo propuesto para la tarea;

- C_{af} - costo de reparo o repuesto y daño colateral (si hay) después de la falla;
- $MTBF$ - MTBF sin tareas preventivas;
- $MTTF_p$ - MTBF de los componentes que fallan antes del intervalo propuesto para la tarea de mantenimiento.

5.4 Control del mantenimiento

El control de una sección de mantenimiento puede ser ejercido por un método presentado por [DECEX (2017)], llamado control de fila de espera (*BACKLOG*). Se define la fila de espera como el tiempo para que un material pueda ser atendido por determinado proceso o, más precisamente, la relación entre la tasa de llegada y la tasa de atendimento por el proceso. En términos de mantenimiento, es el tiempo para que el equipo de mantenimiento concluya todos los servicios restantes, considerando que no haya nuevas solicitudes.

El algoritmo para el análisis del *BACKLOG* es el siguiente:

1. Relacionar los servicios pendientes en el taller en el inicio del periodo de control (semanal, mensual etc);
2. Sumar los tiempos de arreglo para cada Orden de Servicio (OS) abierta, en hombres-hora (Hh);
3. Sumar los tiempos estimados de mantenimiento (MTTR, MTMP) con los tiempos de los servicios en ejecución;
4. Sustraer los tiempos de las OS ejecutadas en el periodo;
5. Dividir el total de los tiempos estimados restantes, después de la sustracción, por los Hh disponibles (volumen de trabajo) del grupo considerado.

■ **Ejemplo 5.11** Sea un equipo con disponibilidad media de 576 Hh por semana. La contabilización del *BACKLOG* empieza con un volumen de trabajo de 8927 Hh, lo que corresponde a un $BACKLOG=16,0$. Eso indica la necesidad de 16 semanas para la conclusión de los servicios pendientes, caso no sean recibidas nuevas OS.

La capacidad de ejecución semanal es variable (en esto caso), pero la disponibilidad media es constante, para fines de cálculo de *BACKLOG*.

La Tabla 5.2 presentará un ejemplo de cálculo del *BACKLOG* de un taller.

Tabla 5.2

Ejemplo de cálculo de BACKLOG

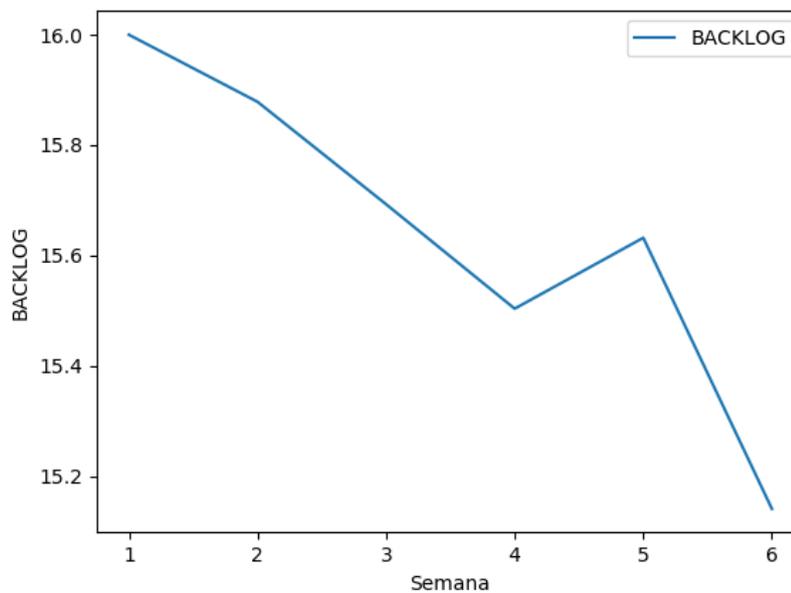
Semana	Inicio (Hh)	Entrada (Hh)	Ejecutada (Hh)	Final (Hh)	BACKLOG
1	8927	772	483	9216	16,00
2	9216	540	610	9146	15,88
3	9146	413	520	9039	15,69
4	9039	390	499	8930	15,50
5	8930	612	538	9004	15,63
6	9004	313	596	8721	15,14

El código 5.4 calculará interactivamente el *BACKLOG* para este ejemplo, y la Fig. 5.8 presentará los resultados del procesamiento.

```

# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
#% %
cap=np.float_(input("Informe la capacidad de trabajo semanal, en Hh\n"))
sem=np.int_(input("Informe el numero de semanas\n"))
vecsem=np.arange(sem)
inivec=np.zeros(sem)
entvec=np.zeros(sem)
ejecvec=np.zeros(sem)
finvec=np.zeros(sem)
backvec=np.zeros(sem)
inivec[0]=input("Informe la pendencia inicial, em Hh\n")
entvec[0]=input("Informe la entrada inicial, em Hh\n")
ejecvec[0]=input("Informe la cantidad ejecutada inicial, em Hh\n")
for i in range (sem):
    if i!=0:
        inivec[i]=finvec[i-1]
        entvec[i]=input("Informe la entrada, em Hh\n")
        ejecvec[i]=input("Informe la cantidad ejecutada, em Hh\n")
        finvec[i]=inivec[i]+entvec[i]-ejecvec[i]
        backvec[i]=finvec[i]/cap
#% %
plt.plot(np.add(vecsem,1),backvec,label="BACKLOG")
plt.xlabel("Semana")
plt.ylabel("BACKLOG")
plt.legend(loc="best")
plt.show()

```

Código 5.4: Cálculo del *BACKLOG*Figura 5.8. Cálculo del *BACKLOG*



6. Aprendizaje de Máquina

En este capítulo final, serán hechas breves consideraciones respecto al aprendizaje de máquina (del inglés *machine learning*) en el contexto del mantenimiento. Sin embargo, el lector que sienta necesidad de aplicaciones más robustas o más complejas para sus problemas específicos deberá buscar la literatura especializada para profundizar sus conocimientos. Las referencias citadas en este libro son consideradas buenas líneas de partida para investigaciones en esta rama del conocimiento.

La motivación para este tema es la aplicación del aprendizaje de máquina en la Industria 4.0, o también conocida como la Cuarta Revolución Industrial.

La Industria 4.0 es caracterizada por [ABDI (2019)] como un conjunto de tecnologías que permiten la fusión del mundo físico, digital y biológico. En la Industria 4.0 son reunidas tecnologías de automatización y tráfico de datos, siendo frecuentemente asociada al internet de las cosas (del inglés, *internet of things (IoT)*) y al almacenamiento de datos en nube.

Tratándose de máquinas que hacen el tráfico de datos en tiempo real, un sistema de monitoreo de la salud de la máquina puede ser implementado para que sean verificadas tendencias en su desempeño. Un nivel de vibraciones creciente, un aumento en la temperatura de una parte importante de la máquina, un ruido que sea detectado por micrófonos como una anomalía, todo puede ser acompañado visando la búsqueda de evoluciones en esos parámetros. Así, al ser verificada una tendencia de un parámetro que sea desfavorable al funcionamiento del sistema, una acción inmediata, programada en la computadora, podrá ser aplicada. Esa acción además de identificar una anomalía, permitirá indicar el origen del daño, una vez que ciertos niveles en los datos pueden ser característicos de un subsistema específico. Por ejemplo, un desequilibrio en un eje rotativo puede ser causado por un rodamiento dañado, que impone al sistema una frecuencia de vibración característica o una temperatura en región cercana más elevada que un valor normal y conocido.

[Coppin (2015)] cita que aprendizaje de máquina es un segmento en la Inteligencia Artificial que posee la capacidad de aprender conceptos y de generalizar un conjunto de datos de entrenamiento para que sea posible clasificar correctamente datos que no fueron observados anteriormente.

6.1 Nociones básicas

El aprendizaje de máquina, en una forma resumida, es basado en 3 niveles de *feedback* que pueden ser clasificados de la siguiente manera [Russell y Norvig (2013)]:

- **Aprendizaje no supervisado:** hay ciertos patrones en la entrada de datos, pero no hay ningún *feedback* específico. Entonces, en este tipo de aprendizaje, de los agrupamientos (*clustering* y reducción dimensional) se originan los algoritmos más comunes. Por ejemplo, un agente computacional puede definir el funcionamiento de una máquina como “bueno” o “malo” sin nunca haber sido informado a la computadora ejemplos de esos comportamientos;
- **Aprendizaje por refuerzo:** es un aprendizaje en que el agente recibe recompensas o punitivos por sus resultados buenos o malos, respectivamente; y
- **Aprendizaje supervisado:** el agente observa los datos de entrada y salida para un conjunto destinado al aprendizaje. Luego, establece un mapeo de los datos de la entrada para la salida. Entonces, genera una función que se aproxima al conjunto de datos, llamada espacio de hipótesis. En este tipo de aprendizaje, los principales algoritmos son el de clasificación de datos y de regresión.

6.1.1 Clasificación de parámetros de sistema

Modelo simple de clasificación

■ **Ejemplo 6.1** Sea un equipamiento que después de estudiado por un largo periodo, presentó las siguientes características:

- Si funciona bien, su amplitud media de vibración sigue una distribución normal $N(4; 0, 25)$ mm y su temperatura es de $N(60; 4)$ °C;
- Si se encuentra en estado de falla, hay aumento de vibración y de temperatura, siendo su amplitud media de vibración una distribución normal $N(7; 0, 81)$ mm y su temperatura es de $N(85; 25)$ °C;

El código 6.1 realizará una simulación de 50 puntos en las dos condiciones citadas, como si fuesen datos colectados de máquinas reales. También recibirá la información de que si el conjunto de datos pertenece a la primera condición (máquina funcionando bien), entonces recibe un rótulo 1. Si los datos vienen del conjunto de datos perteneciente a una máquina en estado de falla, recibe un rótulo 2.

Después, los datos serán procesados con la biblioteca Scikit Learn, con el método *support vector classifier (SVC)*.

Una vez que el sistema aprendió las reglas de clasificación, serán simulados 15 puntos al azar, pertenecientes a la clasificación 1 o 2. El sistema identificará, por sí sólo, si cada dato simulado deberá recibir el rótulo 1 o 2. Entonces, marcará el dato en la región correspondiente de la gráfica, según una simbología diferenciada.

En una tomada de datos en tiempo real, la clasificación serviría como referencia. La identificación de los puntos siendo de una o de otra clase podría servir para indicar normalidad del sistema o para hacer accionar una alarma, indicando la necesidad de mantenimiento.

Este tipo de clasificación puede asumir datos multidimensionales, es decir, m parámetros y n clases.

```

# -*- coding: utf-8 -*-
import matplotlib.pyplot as plt
import numpy as np
from sklearn.svm import SVC
#%%
n=50 #numero de muestras para cada condicion
mula=4 #media de la amplitude de vibracion del sistema sin fallas
sigla=0.5 #desviacion estandar de la amplitude de vibracion del sistema sin
fallas
mult=60 #media de la temperatura del sistema sin fallas
siglt=2 #desviacion estandar de la temperatura del sistema sin fallas
mu2a=7 #media de la amplitude de vibracion del sistema con fallas
sig2a=0.9 #desviacion estandar de la amplitude de vibracion del sistema con
fallas
mu2t=85 #media de la temperatura del sistema con fallas
sig2t=5 #desviacion estandar de la temperatura del sistema con fallas
#%%Simulacion de datos de referencia
a1=np.random.normal(loc=mula, scale=sigla, size=n)
t1=np.random.normal(loc=mult, scale=siglt, size=n)
at1=np.asarray(list(zip(a1, t1)))
l1=np.ones(n)
a2=np.random.normal(loc=mu2a, scale=sig2a, size=n)
t2=np.random.normal(loc=mu2t, scale=sig2t, size=n)
at2=np.asarray(list(zip(a2, t2)))
l2=2*np.ones(n)
at=np.concatenate((at1, at2), axis=0)
l=np.concatenate((l1, l2), axis=0)
#%%Simulacion de datos para predecir
n1=15
a3=np.zeros(n1)
t3=np.zeros(n1)
for i in range(n1):
    if np.random.randint(2)==0:
        a3[i]=np.random.normal(loc=mula, scale=sigla)
        t3[i]=np.random.normal(loc=mult, scale=siglt)
    else:
        a3[i]=np.random.normal(loc=mu2a, scale=sig2a)
        t3[i]=np.random.normal(loc=mu2t, scale=sig2t)
at3=np.asarray(list(zip(a3, t3)))
#%%Clasificacion de los datos
clf=SVC(kernel='linear')
clf.fit(at, l)
l3=clf.predict(at3)
#%%
plt.scatter(t1, a1, color='g', label="Normal")
plt.scatter(t2, a2, color='r', label="Falla")
for i in range(n1):
    if l3[i]==1:
        plt.scatter(t3[i], a3[i], color='k', marker='^')
    else:
        plt.scatter(t3[i], a3[i], color='r', marker='x')
plt.legend(loc="best")
plt.xlabel("Temperatura")
plt.ylabel("Amplitud media de vibracion")
plt.title("Clasificador de sistema")
plt.show()

```

Código 6.1: Clasificación de parámetros de sistema

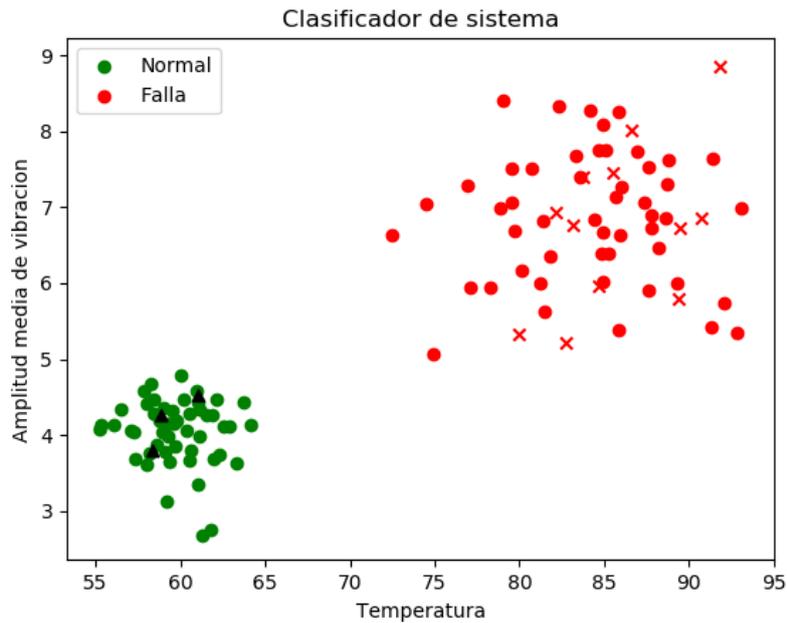


Figura 6.1. Ejemplo de un clasificador de parámetros



K-means

Otra manera de clasificar datos agrupados es por medio de los *K-means*. Esta es una técnica de *clustering* que permite establecer rótulos al agrupamiento de datos.

■ **Ejemplo 6.2** Sea un equipo rotativo que es monitoreado en frecuencia y en amplitud de vibración. Esos dos parámetros son característicos de subsistemas del equipo que, aunque acoplados, giran y oscilan con intensidades distintas.

Entonces, un ingeniero trabajando en mantenimiento puede clasificar ese conjunto de datos con rótulos (en el caso de este ejemplo, con colores) e intentar identificar datos anómalos que huyan de los patrones conocidos, lo que sería posible accionar una alarma para interrumpir la operación de inmediato y permitir el mantenimiento antes que daños mayores ocurran.

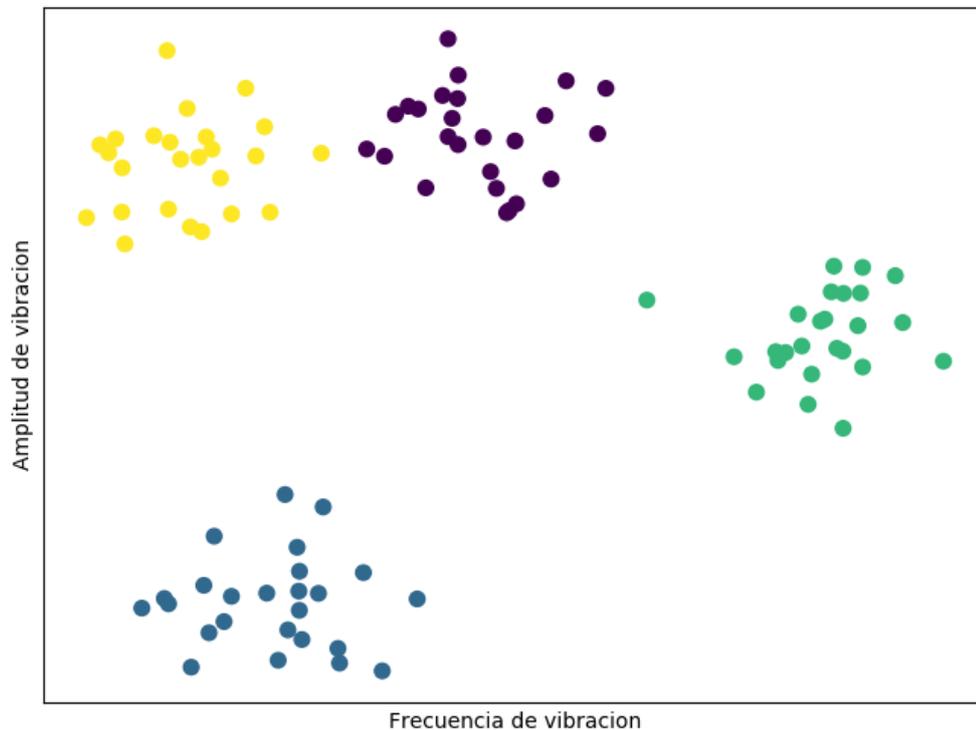
Este tipo de clasificación también puede asumir datos multidimensionales, es decir, m parámetros y n clases.

En el código 6.2, serán generados 100 puntos al azar, para que sean automáticamente clasificados por la computadora. En realidad, los datos vendrían de sensores multicanales y serían agrupados en una misma pantalla para su análisis.

```

# -*- coding: utf-8 -*-
from sklearn.datasets.samples_generator import make_blobs
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
#%%
X, y = make_blobs(n_samples=100, centers=4, random_state=42, cluster_std
                 =1.5)
model = KMeans(4, random_state=0)
y = model.fit_predict(X)
#%%
fig, ax = plt.subplots(figsize=(8, 6))
ax.scatter(X[:, 0], X[:, 1], s=50, c=y, cmap='viridis')
plt.tick_params(top='off', bottom='off', left='off', right='off', labelleft='off',
               labelbottom='off')
plt.xlabel("Frecuencia de vibracion")
plt.ylabel("Amplitud de vibracion")
plt.show()

```

Código 6.2: Clustering de parámetros por *K-means*Figura 6.2. Ejemplo de un clasificador de parámetros por *K-means*

Aunque los datos parezcan artificialmente rotulados por colores, sus agrupamientos fueron matemáticamente establecidos en las líneas 7 y 8 del código 6.2.



Árbol de decisión

Un árbol de decisión es la representación de una función que recibe valores de entrada en la forma de un vector y tiene como salida una decisión. Los valores de entrada pueden ser continuos o discretos, incluso booleanos.

La Fig. 6.3 ilustrará un árbol de decisión.

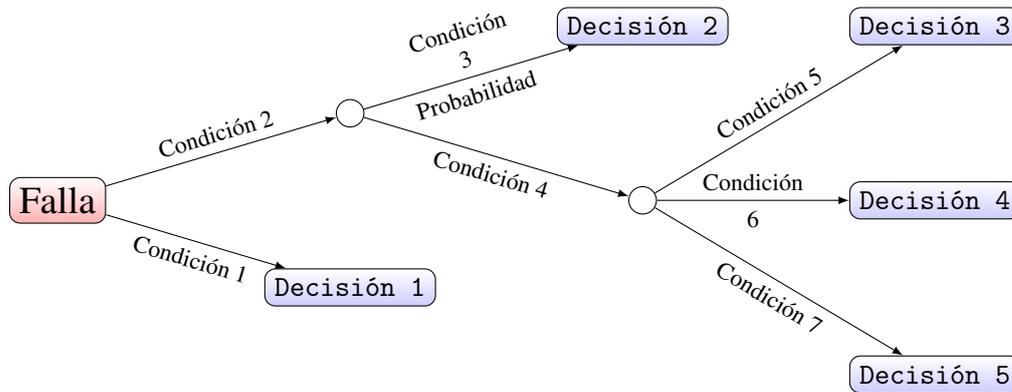


Figura 6.3. Ejemplo de árbol de decisión

La teoría matemática de soporte a los árboles de decisión puede ser accedida en la literatura técnica de Inteligencia Artificial, como en [Russell y Norvig (2013)] y en [Coppin (2015)]. En este capítulo, será presentada una técnica, basada en la biblioteca Scikit Learn, que clasifica datos bidimensionales por medio de árboles de decisión.

■ **Ejemplo 6.3** En este ejemplo, será generado un árbol de decisión respecto a la falla de un carro. Serán combinadas características sencillas, que pueden originar una falla o no. La salida será booleana, pero podría ser continua o discreta, esta con múltiples valores.

La siguiente tabla presentará parámetros de los subsistemas del carro. Serán presentadas 10 condiciones e informadas a la computadora si originan o no una falla. Los niveles paramétricos serán informados como alto, medio, bajo o nulo, pero también podrían ser definidos como variables continuas.

Tabla 6.1

Árbol de fallas de un carro

	Combustible	Temperatura del agua	Nivel de la batería	Falla
0	Medio	Baja	Bajo	No
1	Alto	Media	Bajo	No
2	Alto	Alta	Alto	Si
3	Bajo	Media	Nulo	Si
4	Nulo	Baja	Nulo	Si
5	Alto	Baja	Nulo	Si
6	Bajo	Baja	Bajo	No
7	Alto	Media	Bajo	No
8	Alto	Baja	Alto	No
9	Bajo	Media	Alto	No

Partiendo de los datos arriba, será construido un *dataframe* con la biblioteca Pandas.

Entonces, será necesario calcular la entropía del sistema, por medio de la ecuación de Shannon.

$$S = -\sum_{i=1}^N p_i \log_2 p_i \quad (6.1)$$

Una vez calculada la entropía para cada combinación de condiciones de los subsistemas, es calculada la ganancia de información (IG) para cada atributo. La IG es definida por [Coppin (2015)] como una reducción en entropía (Ec. (6.2)). La intención es determinar cuál atributo presenta la mayor IG, lo que será considerado el nodo vencedor. En otras palabras, el nodo vencedor tendrá el mayor nivel de ganancia de información.

$$S = -p_1 \log_2 p_1 - p_0 \log_2 p_0 \quad (6.2)$$

En la Ec. (6.2), p_1 es definido como la proporción de los datos de entrenamiento que incluye ejemplos positivos y p_0 los correspondientes a los ejemplos negativos. En la Tab. 6.1, por ejemplo, hubo 4 casos de fallas (para nuestro interés, observaciones positivas) y 6 casos de funcionamiento normal (observaciones negativas). Luego, la entropía para ese conjunto de datos es $S = -(4/10)\log_2(4/10) - (6/10)\log_2(6/10) = 0,9710$. La entropía sería exactamente 1 si los éxitos y los fracasos estuviesen en igual cantidad.

A continuación, se calcula la entropía para cada subconjunto de datos (ejemplo: combustible alto que generó éxito más combustible alto que generó fracaso; y así sucesivamente). Enseguida, se hace la suma de la entropía original (0,9710) con todas las entropías de los subconjuntos, para cada atributo. El atributo que tiene la mayor ganancia de información es posicionado en el topo del árbol y entonces los demás subsecuentes componen las ramas del árbol.

```

# -*- coding: utf-8 -*-
import numpy as np
import pandas as pd
eps = np.finfo(float).eps #Numero mucho pequeno, diferente de cero
from numpy import log2 as log
#%
dataset={'Combustible':['Medio','Alto','Alto','Bajo','Nulo','Alto','Bajo','
Alto','Alto','Bajo'],'Temp agua':['Bajo','Medio','Alto','Medio','Bajo','
Bajo','Bajo','Medio','Bajo','Medio'],'Nivel Bat':['Bajo','Bajo','Alto','
Nulo','Nulo','Nulo','Medio','Bajo','Alto','Alto'],'Falla':['No','No','Si'
,'Si','Si','Si','No','No','No','No']}
df = pd.DataFrame(dataset ,columns=['Combustible','Temp agua','Nivel Bat','
Falla'])
#%
def busq_entropia(df):
    Class = df.keys()[-1]
    entropia = 0
    valores = df[Class].unique()
    for valor in valores:
        fraccion = df[Class].value_counts()[valor]/len(df[Class])
        entropia += -fraccion*log(fraccion)
    return entropia
def busq_entropia_atributo(df, atributo):
    Class = df.keys()[-1]
    variables_blanco = df[Class].unique()
    variables = df[atributo].unique()
    entropia2 = 0
    for variable in variables:
        entropia = 0
        for variable_blanco in variables_blanco:
            num = len(df[atributo][df[atributo]==variable][df[Class] ==
variable_blanco])
            den = len(df[atributo][df[atributo]==variable])
            fraccion = num/(den+eps)
            entropia += -fraccion*log(fraccion+eps)
        fraction2 = den/len(df)
        entropia2 += -fraction2*entropia
    return abs(entropia2)
def busq_vencedor(df):
    IG = []
    for key in df.keys()[:-1]:
        IG.append(busq_entropia(df)-busq_entropia_atributo(df, key))
    return df.keys()[:-1][np.argmax(IG)]
def obt_subtabla(df, node, value):
    return df[df[node] == value].reset_index(drop=True)
def constArbol(df, arbol=None):
    node = busq_vencedor(df)
    atrValor = np.unique(df[node])
    if arbol is None:
        arbol={}
        arbol[node] = {}
    for value in atrValor:
        subtabla = obt_subtabla(df, node, value)
        clValor, counts = np.unique(subtabla['Falla'], return_counts=True)
        if len(counts)==1:
            arbol[node][value] = clValor[0]
        else:
            arbol[node][value] = constArbol(subtabla)
    return arbol
def predic(inst, arbol):

```

```

    for nodes in arbol.keys():
        value = inst[nodes]
        arbol = arbol[nodes][value]
        prediccion = 0
        if type(arbol) is dict:
            prediccion = predic(inst, arbol)
        else:
            prediccion = arbol
            break;
    return prediccion
%%Hacer una prediccion con una combinacion conocida
inst=df.iloc[6]
arbol=constArbol(df)
prediccion=predic(inst, arbol)
prediccion
%%Hacer una prediccion de una condicion al azar
def atrib(i):
    if i==0:
        atb='Nulo'
    elif i==1:
        atb='Bajo'
    elif i==2:
        atb='Medio'
    else:
        atb='Alto'
    return atb
cond={'Combustible': atrib(np.random.randint(4)), 'Temp agua': atrib(np.random
    randint(4)), 'Nivel Bat': atrib(np.random.randint(4))}
inst=pd.Series(cond)
prediccion=predic(inst, arbol)
prediccion

```

Código 6.3: Inferencia del estado de un sistema basado en parámetros de sus subsistemas

El código tiene la capacidad de que luego después de que hayan sido entrenados los datos, hacer una predicción a partir de condiciones de falla conocidas (Tab. 6.1) y generar condiciones de funcionamiento al azar, con la finalidad de hacer predicciones con datos simulados. Sin embargo, este es un código no determinista, que posibilita un nivel de aciertos grande pero no inequívoco.

Es importante destacar que, en este ejemplo, los parámetros Combustible y Nivel de Batería son sometidos a condición de falla cuando son valores nulos. Pero la temperatura del agua presenta una condición de falla cuando su valor es alto.

El código 6.4 presentará los mismos cálculos, pero usando la biblioteca Scikit Learn. En este caso, los valores simbólicos [*nulo, bajo, medio, alto*] fueron sustituidos por los valores numéricos [0, 1, 2, 3], bien como los valores booleanos [*nofalla, falla*] por [0, 1].

```

# -*- coding: utf-8 -*-
import numpy as np
import pandas as pd
from sklearn import tree
#%%
dataset={'Combustible':[2,3,3,1,0,3,1,3,3,1], 'Temp agua'
        :[1,2,3,2,1,1,1,2,1,2], 'Nivel Bat':[1,1,3,0,0,3,2,1,3,3], 'Falla'
        :[0,0,1,1,1,1,1,0,0,0]}
datasetaux={'Combustible':[2,3,3,1,0,3,1,3,3,1], 'Temp agua'
           :[1,2,3,2,1,1,1,2,1,2], 'Nivel Bat':[1,1,3,0,0,3,2,1,3,3], 'Falla'
           :[0,0,1,1,1,1,1,0,0,0]}
df = pd.DataFrame(dataset, columns=['Combustible', 'Temp agua', 'Nivel Bat', '
Falla'])
#%%
y=datasetaux.pop("Falla")
x=datasetaux
clf = tree.DecisionTreeClassifier()
clf = clf.fit(np.column_stack((x.get("Combustible"), x.get("Temp agua"), x.
get("Nivel Bat"))), y)
#%%Hacer una prediccion con una combinacion conocida
inst=np.zeros(3)
evento=2
for i in range (3):
    inst[i]=list(x.values())[i][evento]
if clf.predict([inst])[0]==1:
    fal="falla"
else:
    fal="no falla"
print("Si el sistema recibe las condiciones Combustible=%d, Temperatura del
agua=%d y Nivel de bateria=%d, entonces %s" %(inst[0], inst[1], inst[2], fal
))
#%%Hacer una prediccion de una condicion al azar
inst=np.random.randint(4, size=3)
if clf.predict([inst])[0]==1:
    fal="falla"
else:
    fal="no falla"
print("Si el sistema recibe las condiciones Combustible=%d, Temperatura del
agua=%d y Nivel de bateria=%d, entonces %s" %(inst[0], inst[1], inst[2], fal
))

```

Código 6.4: Inferencia del estado de un sistema basado en parámetros de sus subsistemas



Matriz de confusión

Una matriz de confusión permite establecer la calidad de la salida de los códigos no deterministas, comparando los valores predichos con los valores verdaderos.

La diagonal principal de una matriz de confusión representa el número de predicciones que están en acuerdo con el valor real. Cuanto mayores sean los números en la diagonal principal, mejor la capacidad de predicción.

Por ejemplo, sea un sistema que predice condiciones de falla o de no falla, a partir de datos de características de sus atributos, como en los ejemplos de la sección anterior. Si el test del sistema hace n operaciones, es posible hacerse la clasificación de los resultados como: verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN).

Tabla 6.2

Matriz de confusión para el código 6.4

	Predicho: no falla	Predicho: falla
Real: no falla	TN	FP
Real: falla	FN	TP

■ **Ejemplo 6.4** En el código 6.5, serán generadas 50 combinaciones al azar con el código 6.4 y los resultados serán analizados por el operador. Enseguida, será montada una matriz de confusión con las salidas del código, con el fin de evaluar su eficiencia.

```

# -*- coding: utf-8 -*-
import numpy as np
import pandas as pd
from sklearn import tree
# %%
dataset={ 'Combustible':[2,3,3,1,0,3,1,3,3,1], 'Temp agua'
          :[1,2,3,2,1,1,1,2,1,2], 'Nivel Bat':[1,1,3,0,0,3,2,1,3,3], 'Falla'
          :[0,0,1,1,1,1,0,0,0,0]}
datasetaux={ 'Combustible':[2,3,3,1,0,3,1,3,3,1], 'Temp agua'
            :[1,2,3,2,1,1,1,2,1,2], 'Nivel Bat':[1,1,3,0,0,3,2,1,3,3], 'Falla'
            :[0,0,1,1,1,1,0,0,0,0]}
df = pd.DataFrame(dataset , columns=[ 'Combustible' , 'Temp agua' , 'Nivel Bat' , '
    Falla' ])
# %%
y=datasetaux .pop(" Falla")
x=datasetaux
clf = tree .DecisionTreeClassifier()
clf = clf .fit(np .column_stack((x .get(" Combustible" ) , x .get(" Temp agua" ) ,x .
    get(" Nivel Bat" ))), y)
# %%Hacer predicciones al azar
n=50
evalu=np .zeros([n,4])
for i in range (n):
    inst=np .random .randint(4 , size=3)
    for j in range (4):
        if j <=2:
            evalu[i][j]=inst[j]
        else:
            evalu[i][j]=clf .predict([ inst ])[0]
# %%
tn=int(input("Informe el numero de no fallas que fueron predichas
correctamente\n"))
tp=int(input("Informe el numero de fallas que fueron predichas correctamente\n"))
fn=int(input("Informe el numero de no fallas que no fueron predichas
correctamente\n"))
fp=int(input("Informe el numero de fallas que no fueron predichas
correctamente\n"))
matconf={ 'Predicho No Falla':[ tn ,fp ] , 'Predicho Falla':[ fn , tp ]}
dfmatconf=pd .DataFrame(matconf , index=[ 'Real No Falla' , 'Real Falla' ])

```

Código 6.5: Matriz de confusión para el código 6.4

Index	Predicho No Falla	Predicho Falla
Real No Falla	20	0
Real Falla	7	23

Figura 6.4. Ejemplo de una matriz de confusión

Se observa en la Fig. 6.4 que el número de predicciones correctas fue elevado (86%), en cuanto se observaron 7 falsos negativos (14%). De un análisis de la causa del gran número de FP, fue posible concluir que todos los falsos negativos se deben a la falla de combustible, cuando este se presentó nulo, al azar. Se puede observar en la Tabla 6.1 que la única entrada en donde el combustible es nulo es la entrada de la línea 4. Pero esa misma entrada posee una otra entrada nula, correspondiente al nivel de carga de la batería. Entonces, la decisión no fue buena, porque hubo dos informaciones que causaban la falla y el espacio de muestras es pequeño para concluirse que el combustible nulo también causa fallas. El código es dependiente de la buena calidad y de la abundancia de datos, como fue citado en la Sección 2.3.

■

6.1.2 Redes neuronales artificiales

La base de una red neuronal artificial es el funcionamiento de las neuronas biológicas. En resumen, cuando una combinación lineal de las conexiones de entrada (Fig. 6.5) supera un valor límite, entonces es activada una función que permite una salida de datos para una o para diversas conexiones, que pueden ser conectadas a otras neuronas.

Las redes neuronales poseen nodos conectados que propagan las activaciones a_i para a_j . Las conexiones tienen un peso numérico $w_{i,j}$ que establece la fuerza y la señal de la conexión.

El desarrollo matemático de las diversas redes neuronales artificiales puede ser accedido en la literatura especializada.

Las funciones de activación más comunes son el escalón y la sigmoidea.

El código 6.6 y la Fig. 6.6 ilustrarán dos ejemplos de funciones de activación.

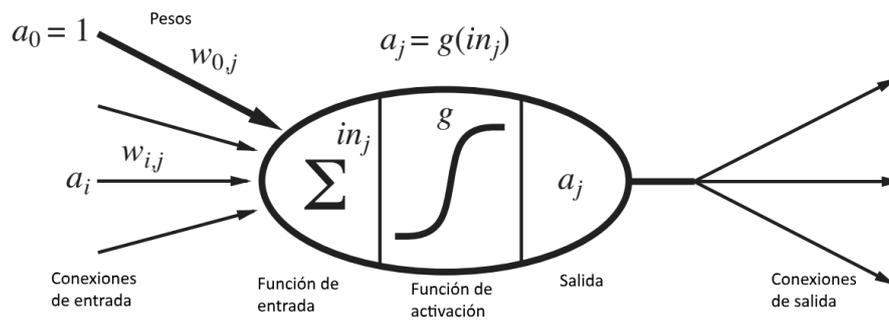


Figura 6.5. Ejemplo de una neurona. Copyright 2013 por Elsevier

```

# -*- coding: utf-8 -*-
import numpy as np
import scipy.special as sp
import matplotlib.pyplot as plt
#%%
x=np.linspace(-6,6)
y1=np.zeros(len(x))
y2=np.zeros(len(x))
for i in range(len(x)):
    if x[i]<=0:
        y1[i]=0
    else:
        y1[i]=1
    y2[i]=sp.expit(x[i])
#%%
plt.plot(x,y1,label="Escalon")
plt.plot(x,y2,label="Sigmoidea")
plt.title("Funciones de activacion")
plt.legend(loc="best")
plt.show()

```

Código 6.6: Generación de funciones de activación

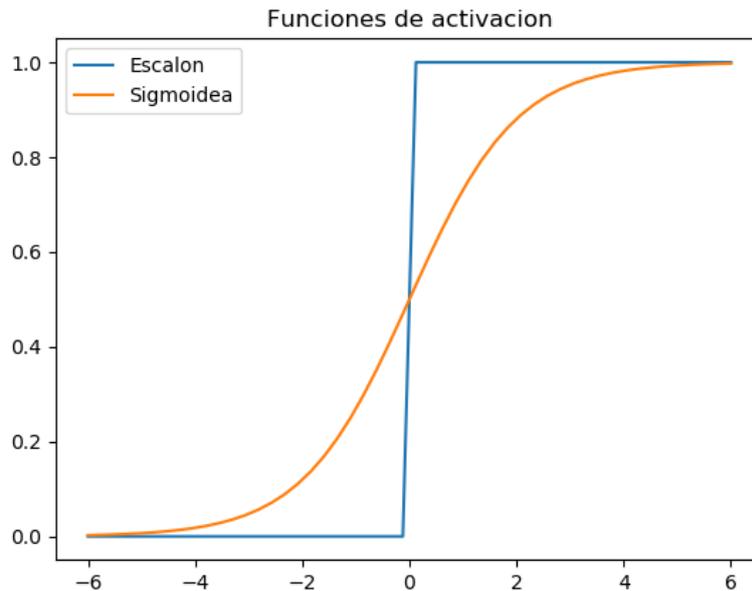


Figura 6.6. Ejemplos de funciones de activación

La Fig. 6.7 ilustrará diversos tipos de redes neuronales artificiales.

■ **Ejemplo 6.5** En este ejemplo, será mostrado el funcionamiento de una neurona artificial, que es activada por una función sigmoidea y tiene los siguientes parámetros: $w=[0,1]$ y $b=4$, como vector de pesos y valor de *bias*, respectivamente.

La técnica será de alimentación para adelante.

La pregunta del ejemplo es: si se entra con un vector $x=[2,3]$, ¿cuál será la salida de la neurona, empleándose el perceptron sigmoideo?

Analíticamente, el cálculo es el siguiente:

$$(w \cdot x) + b = ((w_1 x_1) + (w_2 x_2)) + b = 0 * 2 + 1 * 3 + 4 = 7 \quad (6.3)$$

El potencial de activación fue calculado como 7. En seguida, se aplica la función sigmoidea a ese valor.

$$y = f(w \cdot x + b) = f(7) = \frac{1}{1 + e^{-7}} = 0,999 \quad (6.4)$$

La neurona responderá como salida el valor de 0,999.

■

Uno de los objetivos de la aplicación de las redes neuronales artificiales, en mantenimiento, es conocer una herramienta de predicción de la evolución de un determinado parámetro de control de una máquina en el tiempo. Una vez que la evolución puede ser no determinista, incluso caótica, en la ausencia de modelos deterministas, un modelo por redes neuronales artificiales puede solventar la necesidad de estimar valores cercanos a los límites de conocimiento de una serie temporal o para detectarse anomalías.

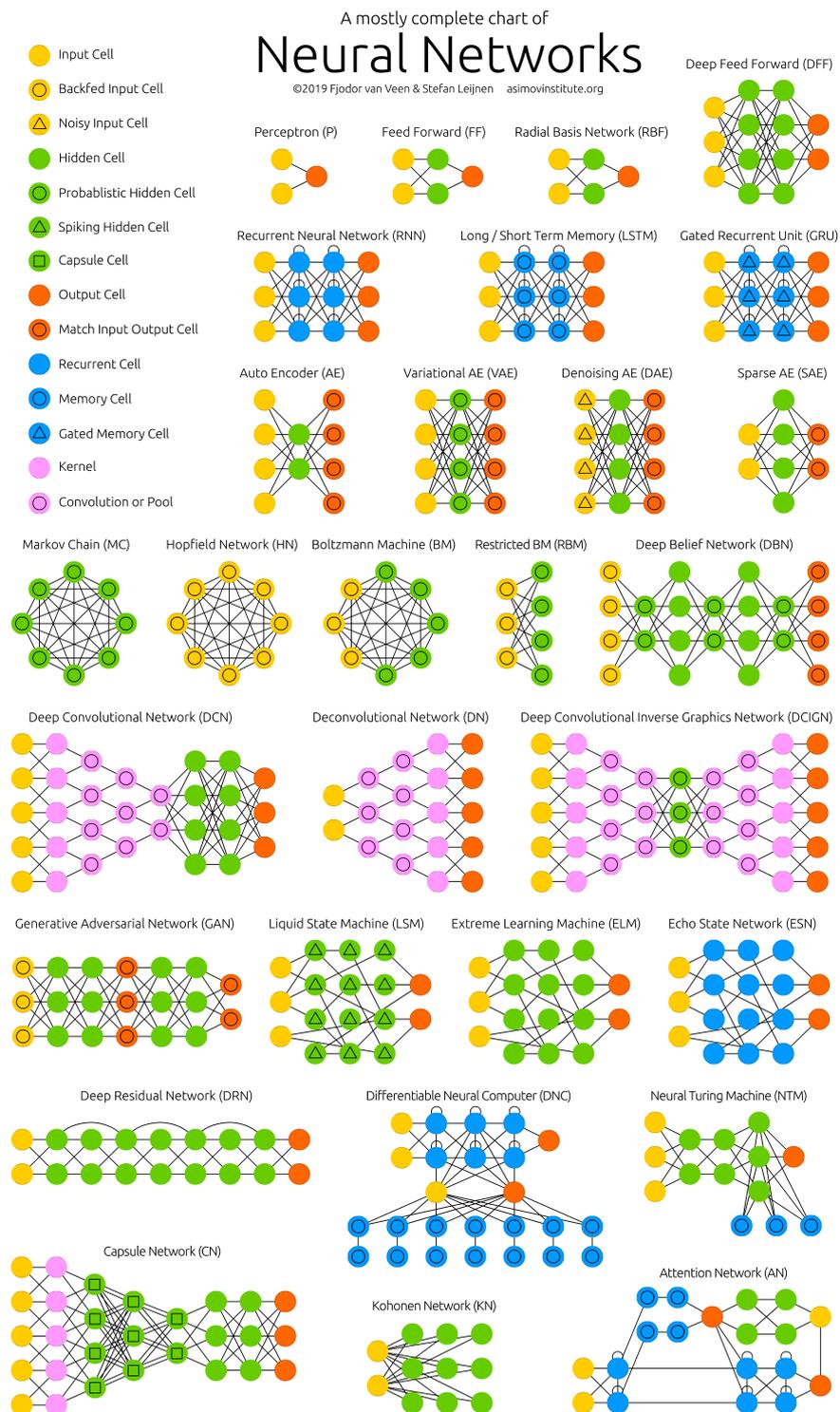


Figura 6.7. Ejemplos de redes neuronales artificiales. Copyright 2017 por The Asimov Institute

■ **Ejemplo 6.6** En el código 6.7 será presentada una predicción de un valor de una serie temporal simple, a modo que ilustración. Será usada la biblioteca Keras y la técnica será un modelo del tipo perceptron multicamadas (MLP), con una camada oculta. Una serie temporal $x = [10, 20, \dots, 60, 70]$ será cargada en el código y se desea estimar cual es el valor siguiente que, obviamente, será 80. La función de activación será la *Rectified Linear Unit Activation Function (RELU)* y el número de épocas puede ser comprendido como el número de *loopings* que el código ejecutará para hacer muestras y entrenar las salidas, actualizando los pesos de la red neuronal.

Todos los pesos del modelo serán inicializados con valores al azar y muy pequeños. La actualización de los pesos del modelo será hecha por un proceso decreciente (gradiente) de los errores obtenidos entre los valores de entrenamiento, dados los valores de entrada y los valores conocidos de salida, hasta el valor máximo de *loopings* (2000) o el valor mínimo de error determinado como aceptable.

La actualización de los pesos será calculada por la siguiente ecuación:

$$w \leftarrow w - \eta \frac{\partial E}{\partial w} \quad (6.5)$$

En la Ec. (6.5), el valor negativo indica que si el gradiente muestra el aumento del error respecto al peso, entonces es necesario buscar el menor error posible, tomando el camino opuesto al del gradiente. El parámetro η es una tasa de aprendizaje, que es un tamaño de paso de corrección del peso. Para ese cálculo, será utilizado el método de optimización Adam, combinado con el error medio cuadrático (MSE).

En este caso, dadas las entradas y conocidas las salidas, serán realizadas operaciones en *looping* que buscarán actualizar los pesos, de modo que el modelo de red neural artificial sea adecuado a los datos informados. El modelo utilizará neuronas que recibirán 3 entradas (por ejemplo, [10,20,30]) y retornarán una salida (en el ejemplo, y=40).

```
# -*- coding: utf-8 -*- 1
from numpy import array 2
from keras.models import Sequential 3
from keras.layers import Dense 4
# %% 5
X = array([[10, 20, 30], [20, 30, 40], [30, 40, 50], [40, 50, 60]]) 6
y = array([40, 50, 60, 70]) 7
model = Sequential() 8
model.add(Dense(100, activation='relu', input_dim=3)) 9
model.add(Dense(1)) 10
model.compile(optimizer='adam', loss='mse') 11
model.fit(X, y, epochs=2000, verbose=0) 12
x_input = array([50, 60, 70]) 13
x_input = x_input.reshape((1, 3)) 14
yhat = model.predict(x_input, verbose=0) 15
print(yhat) 16
```

Código 6.7: Previsión de un valor en una serie temporal

Entrenando el modelo con 4 conjuntos de entradas y salidas, fue posible estimar los valores de los pesos de las neuronas. El valor retornado por el código fue $yhat=80,199776$, en buen acuerdo con el valor esperado.

■

Red Long Short-Term Memory (LSTM)

Una red recurrente del tipo LSTM hace el entrenamiento de las neuronas con la propagación para atrás. Es usada para la regresión de datos.

La LSTM permite establecer una predicción de los datos de una serie temporal, por ejemplo. Es decir, si se tiene informaciones hasta el tiempo t , ¿cuál sería el valor de la serie en el tiempo $t+1$?

Para ilustrar como ese tipo de predicción puede ser empleada en las previsiones de parámetros en el mantenimiento, será presentado un ejemplo de un sistema dinámico no lineal, sometido a comportamiento caótico, lo cual tiene una previsibilidad determinista muy sensible a las condiciones iniciales. Mismo delante las irregularidades, el modelo por LSTM permite hacer buenas predicciones al largo del tiempo. No es el propósito de este libro entrar en los detalles técnicos del sistema dinámico o del código, sino presentar a título de ilustración la capacidad de predicción de la red LSTM delante una serie temporal.

■ **Ejemplo 6.7** Sea el sistema dinámico no lineal, con actuador basado en material con memoria de forma, descrito por [Savi (2006)] como un polinomio de quinto grado en desplazamiento. [Savi y Pacheco (2002)] presentan sus ecuaciones de movimiento, según el sistema descrito en la Ec. (6.6):

$$\begin{aligned} y_0' &= y_1 \\ y_1' &= \delta \sin(\bar{\omega}\tau) - \xi y_1 - (\theta - 1)y_0 + \beta y_0^3 - \varepsilon y_0^5 \end{aligned} \quad (6.6)$$

Asumiéndose $\bar{\omega} = 1$, $\xi = 0,1$, $\beta = 1,3e3$, $\varepsilon = 4,7e5$, $\theta = 0,7$ y $\delta = 0,06$, el sistema presenta un comportamiento caótico, cuya irregularidad se puede observar en la Fig. 6.8.

El sistema de ecuaciones fue resuelto por el método de Runge-Kutta de 4^a orden, en un intervalo de 200 unidades de tiempo, para generar la serie de datos. Después, fue aplicado el código en el cual 75% de los puntos de la serie fueron dedicados al entrenamiento de la red neuronal artificial y los demás puntos fueron reservados al test del modelo en LSTM.

Se puede observar en la Fig. 6.8 que el modelo se adecuó a los datos de la serie de test con buena capacidad de predicción, permitiendo la previsión del comportamiento físico del sistema, aunque sin el conocimiento del modelo matemático subyacente. En este caso simulado, un sistema bajo la atención del mantenimiento podría ser acompañado por medio de colecta de sus datos y predicción de su comportamiento. Una vez que el comportamiento predicho se acerque a una evidencia de daño, por predicción se puede interrumpir su funcionamiento para las medidas preventivas o correctivas necesarias.

```
# -*- coding: utf-8 -*-
# %%
import numpy as np
import matplotlib.pyplot as plt
#LSTM
import math
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
# %% Constantes del material
beta=1.3e3
thetaa=1.9
theta=0.7 #temperatura adimensional
ksi=0.1
# %%Runge-Kutta 4a Orden
# Oscilador SMA
# Parametros del sistema
# Valores del paper
OM=1
a=theta-1
```

```

b=-beta 23
c=(beta**2)/(4*(thetaa-1)) 24
f0=0.06 25
#Condiciones Iniciales 26
t0=0 27
u0=0 28
v0=0.01 29
#Condicion final 30
tf=200 31
#Funcion aceleracion del oscilador 32
def vp(t,u,v): 33
    fb=f0*np.sin(OM*t) 34
    return (-ksi*v-a*u+b*u**3-c*u**5+fb) 35
%%Runge-Kutta 4a ordem 36
#Intervalo de integracion 37
fa=10 #frecuencia de muestreo (puntos por segundo) 38
n=fa*tf #Numero de intervalos del intervalo de integracion 39
def rk4(n,tf,t0,u0,v0,f0,OM): 40
    h=(tf-t0)/n 41
    Tr=t0 42
    Ur=u0 43
    Vr=v0 44
    45
    salida=np.zeros((n,4)) 46
    taux1=np.zeros((n,1)) 47
    taux2=np.zeros((n,1)) 48
    taux3=np.zeros((n,1)) 49
    taux4=np.zeros((n,1)) 50
    51
    for i in range(n): 52
        k1=h*Vr 53
        l1=h*vp(Tr,Ur,Vr) 54
        k2=h*(Vr+l1/2) 55
        l2=h*vp(Tr+h/2,Ur+k1/2,Vr+l1/2) 56
        k3=h*(Vr+l2/2) 57
        l3=h*vp(Tr+h/2,Ur+k2/2,Vr+l2/2) 58
        k4=h*(Vr+l3) 59
        l4=h*vp(Tr+h,Ur+k3,Vr+l3) 60
        Ur=Ur+(k1+2*(k2+k3)+k4)/6 61
        Vr=Vr+(l1+2*(l2+l3)+l4)/6 62
        Tr=Tr+h 63
        salida[i]=np.array([Tr,Ur,Vr,f0*np.sin(OM*Tr)]) 64
    65
    for j in range(n): 66
        taux1[j]=salida[j,0] #Tiempo 67
        taux2[j]=salida[j,1] #Posicion 68
        taux3[j]=salida[j,2] #Velocidade 69
        taux4[j]=salida[j,3] #Fuerza 70
    return (taux1,taux2,taux3,taux4) 71
taux1=rk4(n,tf,t0,u0,v0,f0,OM)[0] 72
taux2=rk4(n,tf,t0,u0,v0,f0,OM)[1] 73
taux3=rk4(n,tf,t0,u0,v0,f0,OM)[2] 74
taux4=rk4(n,tf,t0,u0,v0,f0,OM)[3] 75
%%Resultados (T,U) 76
plt.figure() 77
line, = plt.plot(taux1, taux2, '—', linewidth=2) 78
plt.show() 79
%%Resultados (U,V) 80
plt.figure() 81
line, = plt.plot(taux2, taux3, '—', linewidth=2) 82
plt.show() 83

```

```

84  #%%Preparacion de los datos
85  def split_sequence(sequence, n_steps):
86      X, y = list(), list()
87      for i in range(len(sequence)):
88          end_ix = i + n_steps
89          if end_ix > len(sequence)-1:
90              break
91          seq_x, seq_y = sequence[i:end_ix], sequence[end_ix]
92          X.append(seq_x)
93          y.append(seq_y)
94      return np.array(X), np.array(y)
95
96  # Definicion de los datos de entrada
97  pdt=0.75 #Percentual destinado al entrenamiento de la red
98  raw_seq = tau2[0:int(len(tau2)*pdt)]
99  # Numero de pasos
100 n_steps = 3
101 # Dividir en muestras
102 X, y = split_sequence(raw_seq, n_steps)
103 #%%LSTM
104 # Ajuste del arreglo
105 n_features = 1
106 X = X.reshape((X.shape[0], X.shape[1], n_features))
107 # Modelo
108 model = Sequential()
109 model.add(LSTM(50, activation='relu', return_sequences=True, input_shape=(
110     n_steps, n_features)))
111 model.add(LSTM(50, activation='relu'))
112 model.add(Dense(1))
113 model.compile(optimizer='adam', loss='mse')
114 # Ajuste del modelo
115 model.fit(X, y, epochs=200, verbose=0)
116 #%%Prediccion
117 yh=[]
118 for i in range(len(tau2)-3):
119     x_input = np.array([tau2[i], tau2[i+1], tau2[i+2]])
120     x_input = x_input.reshape((1, n_steps, n_features))
121     yhat = model.predict(x_input, verbose=0)
122     yh=np.append(yh,yhat)
123 #%%Generacion de graficos
124 plt.title("LSTM aplicada a un sistema no lineal")
125 plt.plot(tau1,tau2,label="Datos del Sistema")
126 plt.plot(tau1[3:int(len(tau2)*pdt)],yh[0:int(len(tau2)*pdt-3)],'o',label="
127     "Datos del Entrenamiento")
128 plt.plot(tau1[int(len(tau2)*pdt)+1:],yh[int(len(tau2)*pdt-3+1):],'o',
129     label="Datos del Test")
130 plt.xlabel("Tiempo")
131 plt.ylabel("Desplazamiento")
132 plt.legend()
133 plt.show()

```

Código 6.8: Previsión de un valor en una serie temporal



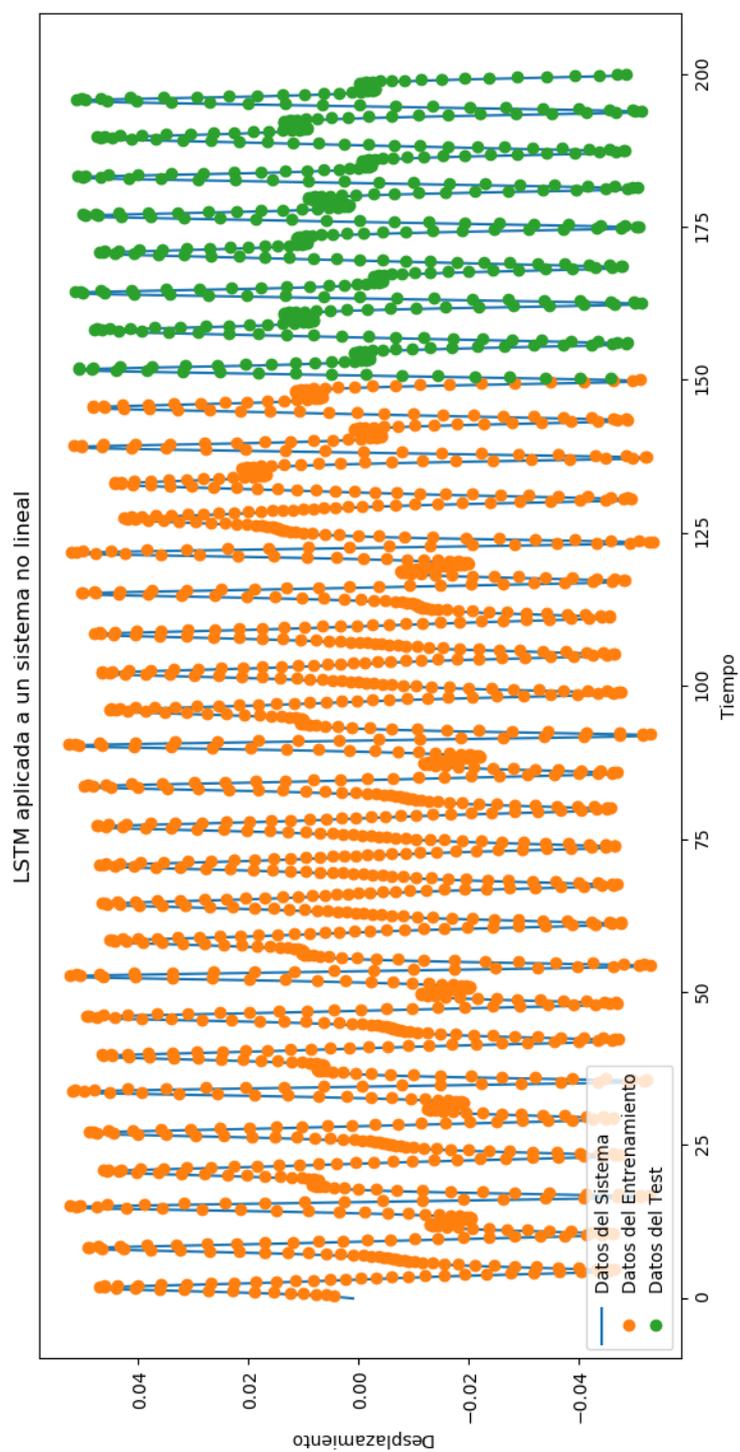


Figura 6.8. Respuesta en posición de un sistema dinámico no lineal



Glosario

autocorrelación Es una correlación de valores de señales con otros valores de la misma señal. Es una medida de aleatoriedad. 56

autorregresión Medida de especificación de que la variable de salida de un modelo depende linealmente de sus valores anteriores bajo un término estocástico. 56

confiabilidad Probabilidad de que un ítem funcione adecuadamente por un período de operación, bajo determinadas condiciones, sin presentar fallas. 8

histograma Es una distribución de frecuencias, representada en gráfico de columnas o de barras, por el cual el conjunto de datos es dividido en clases. Las clases son representadas por la base de cada rectángulo. 38

machine learning Es una forma de Inteligencia Artificial que posibilita al sistema aprender a partir de los datos y no por medio de la programación explícita. 5, 118

mantenimiento Es el conjunto de actividades que son ejecutadas con el objetivo de mantener el material en condiciones de utilización, durante todo su ciclo de vida y, cuando hay averías, restablecer a esa condición. También es conformado por la combinación de acciones técnicas, administrativas y de supervisión, destinadas a mantener o reponer un equipamiento en condiciones de desempeñar, de manera eficaz, las funciones para las que fue diseñado. 5

números pseudoaleatorios Son números que exhiben aleatoriedades estadísticas, aunque sean generados por un proceso determinístico. 40

Python Es un lenguaje de programación de alto nivel, interpretada, de *script*, imperativa, orientada a objetos, funcional, dinámica y multiplataforma. 5

redundancia Un sistema que presenta subsistemas en redundancia indica que cada subsistema en redundancia funciona independientemente de su similar y que el sistema falla cuando todas las redundancias fallan. 71

tasa de fallas Es definida para una población no reparable como la tasa instantánea de fallas para los sobrevivientes hasta el tiempo t durante el próximo instante de tiempo. 8



Lista de Referencias

Referencias bibliográficas

- [ABDI (2019)] Agência Brasileira de Desenvolvimento Industrial (Org.). Indústria 4.0. Disponível en: <<http://www.industria40.gov.br/>>. Acesso en: 15 mayo 2019.
- [ANSI/ASQ Z1.4-2003 (2013)] ASQ, 2013. Sampling Procedures and Tables for Inspection by Attributes (ANSI/ASQ Z1.4-2003 (R2013)), American Society for Quality, Milwaukee, 2013.
- [ANSI/ASQ Z1.9-2003 (2013)] ASQ, 2013. Sampling Procedures and Tables for Inspection by Variables for Percent Nonconforming (ANSI/ASQ Z1.9-2003 (R2013)), American Society for Quality, Milwaukee, 2013.
- [Coppin (2015)] COPPIN, B. Inteligencia Artificial, Primera edición. LTC, 2015.
- [Dhillon (2002)] DHILLON, B. S., 2002. Engineering maintenance: a modern approach. 222 p.
- [DECEX (2017)] Departamento de Educação e Cultura do Exército, Manual de Ensino Gerenciamento da Manutenção, DECEX, Rio de Janeiro, RJ, Junio 2017.
- [DMAVEx (2010)] Diretoria de Material de Aviação do Exército, Modelo de Previsão e Análise do Ciclo de Vida de Material de Defesa Aeronáutico, DMAVEx, Brasília, DF, Julio 2010.
- [DOD (1966)] MIL-HDBK-472, Maintainability Prediction, Department of Defense, Washington D.C., Mayo 1966.
- [DOD (1968)] DOD INST.4151.12, Policies Governing Maintenance Engineering within the Department of Defense, Department of Defense, Washington D.C., Junio 1968.
- [IEEE (2018)] CASS, S., The 2018 Top Programming Languages, IEEE Spectrum, New York, Julio 2018, URL <https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>.
- [Kinnison y Siddiqui (2013)] KINNISON, H. A., SIDDIQUI, T.. Aviation Maintenance Management, Second Edition. McGraw-Hill, 2013.

- [Meyer (1983)] MEYER, P. L., Probabilidade: aplicações à estatística, 2 edição, Rio de Janeiro: Livros Técnicos e Científicos, 1983.
- [Montgomery (2004)] MONTGOMERY, D. C., Introdução ao controle estatístico da qualidade, 4 edição, Rio de Janeiro: Livros Técnicos e Científicos, 2004.
- [NAP (2015)] National Academies Press, Reliability Growth: Enhancing Defense System Reliability, The National Academies of Sciences, Engineering and Medicine, Washington D.C., 2015, Disponible: <https://www.nap.edu/catalog/18987/reliability-growth-enhancing-defense-system-reliability>, Acceso en 5 mayo 2019.
- [NASA (2000)] National Aeronautics and Space Administration, NASA Reliability Centered Maintenance Guide for Facilities and Collateral Equipment, NASA Headquarters, Washington D.C., Febrero 2000.
- [NASA (2002)] National Aeronautics and Space Administration, Fault Tree Handbook with Aerospace Applications, NASA Headquarters, Washington D.C., Agosto 2002.
- [NASA (2004)] National Aeronautics and Space Administration, NASA Reliability Centered Building and Equipment Acceptance Guide, NASA Headquarters, Washington D.C., Julio 2004.
- [NAVAIR (2005)] NAVAIR 00-25-403, Guidelines for the Naval Aviation Reliability-Centered Maintenance Process, Naval Air Systems Command, Patuxent River, Julio 2005.
- [NIST/SEMATECH (2012)] NIST/SEMATECH e-Handbook of Statistical Methods, <http://www.itl.nist.gov/div898/handbook/>, Acceso en 15 abril 2019.
- [Nowlan y Heap (1978)] NOWLAN, F. S., HEAP, F. H.. Reliability-Centered Maintenance. United Airlines and Dolby Press, sponsored and published by the Office of Assistant Secretary of Defense, 1978.
- [PSF (2018)] Python Software Foundation (PSF) (2018), Python: version 3.7.3, Ámsterdam, Países Bajos, URL <https://www.python.org/>
- [RELIASOFT (2015)] ReliaSoft Corporation, Reliability Growth & Repairable System Data Analysis Reference, Arizona, 8 mayo 2015. Disponible: http://reliawiki.org/index.php/Reliability_Growth_and_Repairable_System_Analysis_Reference, Acceso en 5 mayo 2019.
- [Russell y Norvig (2013)] RUSSELL, s. j., NORVIG, P. Inteligência Artificial. Elsevier. 3 edição, 2013.
- [Savi (2006)] SAVI, M. A. Dinâmica não-linear e caos. e-papers. 2006.
- [Savi y Pacheco (2002)] SAVI, M. A., PACHECO, P. M. C. L. Chaos and Hyperchaos in Shape Memory Systems. International Journal of Bifurcation and Chaos. vol. 12, número 3, 2002.
- [Sherif y Kolarik (1981)] SHERIF, Y. S., KOLARIK, W. J.. Life Cycle Costing: Concept and Practice. Omega. vol. 9, número 3, 1978. Disponible: <http://www.sciencedirect.com/science/article/pii/0305048381900359>, Acceso en 23 abril 2019.
- [Stewart (2013)] STEWART, I., 2013. Dezesete equações que mudaram o mundo. Rio de Janeiro: Zahar.

- [Stoco *et al.* (2016)] STOCO, W. H., SANTOS, R. V., LIMA, A., TORREZAN, H. F. Qualidade dimensional: estudos e aplicações do sistema GD&T no processo de desenvolvimento de um produto. In: XXXVI Encontro Nacional de Engenharia de Produção, 10, 2016, João Pessoa, PB. Anais (on-line). Disponível: http://www.abepro.org.br/biblioteca/TN_STO_230_345_28845.pdf Acesso em 15 abril 2019.
- [Vaz y Mansori (2017)] VAZ, A., MANSORI, S. (2017). Target Days versus Actual Days of Finished Goods Inventory in Fast Moving Consumer Goods. *International Business Research*. 10. 19. 10.5539/ibr.v10n6p19.
- [Van Veen (2017)] VAN VEEN, F. Neural Network Zoo Prequel: Cells and Layers. The Asimov Institute Marzo 2017. Disponível: <https://www.asimovinstitute.org/author/fjodorvanveen/>, Acesso em 18 mayo 2019.
- [Vesely (1981)] VESELY, W. E.; et al. Fault Tree Handbook (NUREG-0492). Nuclear Regulatory Commission, 1981. Disponível: <https://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0492/>, Acesso em 7 mayo 2019.
- [Wandeck (2010)] WANDECK, M. GD&T On Line. GD&T Engenharia Ltda. Rio de Janeiro, RJ.

Referencias de las figuras

- National Aeronautics and Space Administration. (2000). Conditional Probability of Failure Curves [Gráfico]. Recuperado de <https://www.hq.nasa.gov/office/codej/codejx/Assets/Docs/RCMGuideMar2000.pdf>.
- Diretoria de Material de Aviação do Exército. (2010). Esquema Ilustrativo dos Ciclos de Vida de um Material [Gráfico]. Recuperado de Modelo de Previsão e Análise do Ciclo de Vida de Material de Defesa Aeronáutico.
- Pergamon Press Ltd. (1981). General phase-cost relationships for a system [Gráfico]. Recuperado de OMEGA The Int. JI of Mgmt Sci. Vol. 9, No. 3. pp. 287 to 296. 1981 0305-0483/81/030287-10502.00/0 Great Britain.
- National Aeronautics and Space Administration. (2004). Life cycle cost [Gráfico]. Recuperado de <https://www.hq.nasa.gov/office/codej/codejx/Assets/Docs/RCB&EGuideJUL04.pdf>.
- National Institute of Standards and Technology. (2012). The Bathtub Curve [Gráfico]. Recuperado de <https://www.itl.nist.gov/div898/handbook/apr/section1/apr124.htm>.
- Livros Técnicos e Científicos. (2004). Ciclos em um gráfico de controle [Gráfico]. Recuperado de Introdução ao Controle Estatístico da Qualidade.
- Livros Técnicos e Científicos. (2004). Um padrão de mistura [Gráfico]. Recuperado de Introdução ao Controle Estatístico da Qualidade.
- Livros Técnicos e Científicos. (2004). Um deslocamento no nível de processo [Gráfico]. Recuperado de Introdução ao Controle Estatístico da Qualidade.

Livros Técnicos e Científicos. (2004). Uma tendência no nível de processo [Gráfico]. Recuperado de Introdução ao Controle Estatístico da Qualidade.

Livros Técnicos e Científicos. (2004). Um padrão de estratificação [Gráfico]. Recuperado de Introdução ao Controle Estatístico da Qualidade.

National Institute of Standards and Technology. (2012). Series Systems Reduced to Equivalent One Component Systems [Gráfico]. Recuperado de <https://www.itl.nist.gov/div898/handbook/apr/section1/apr182.htm>.

National Institute of Standards and Technology. (2012). Parallel System and Equivalent Single Component [Gráfico]. Recuperado de <https://www.itl.nist.gov/div898/handbook/apr/section1/apr183.htm>.

CRC Press. (2002). An $(k+1)$ -unit standby system [Gráfico]. Recuperado de Engineering maintenance: a modern approach.

Mc Graw Hill Education. (2002). Restoration of system perfection [Gráfico]. Recuperado de Aviation Maintenance Management, Second Edition.

Direction of Commander, Naval Air Systems Command. (2005). On Condition Task Considerations [Gráfico]. Recuperado de Guidelines for the Naval Aviation Reliability-Centered Maintenance Process.

National Aeronautics and Space Administration. (2000). Design Improvements through Maintenance Feedback [Gráfico]. Recuperado de <https://www.hq.nasa.gov/office/codej/codejx/Assets/Docs/RCMGuideMar2000.pdf>.

CRC Press. (2002). System life cycle vs. various types of human error [Gráfico]. Recuperado de Engineering maintenance: a modern approach.

CRC Press. (2002). A fault tree [Gráfico]. Recuperado de Engineering maintenance: a modern approach.

Canadian Center of Science and Education. (2017). Average inventory is Q ordered divided by 2 [Gráfico]. Recuperado de International Business Research; Vol. 10, No. 6; 2017 Canada.

Canadian Center of Science and Education. (2017). Economic Order Quantity (EOQ) model [Gráfico]. Recuperado de International Business Research; Vol. 10, No. 6; 2017 Canada.

Diretoria de Material de Aviação do Exército. (2010). Ilustração do Comportamento do Parâmetro de Sensibilidade S_{CR} [Gráfico]. Recuperado de Modelo de Previsão e Análise do Ciclo de Vida de Material de Defesa Aeronáutico.

Elsevier. (2013). Um simples modelo matemático para um neurônio [Gráfico]. Recuperado de Inteligência Artificial.

The Asimov Institute. (2017). The Neural Network Zoo [Gráfico]. Recuperado de <https://www.asimovinstitute.org/author/fjodorvanveen/>.



Agradecimientos

Fazer agradecimentos é uma tarefa árdua, pelo enorme risco de não se fazer menção a alguém que teve uma contribuição direta ou indireta no atingimento de algum objetivo. O processo de alcance de uma meta – e a elaboração deste livro foi uma meta pessoal que particularmente considero significativa – envolveu muitas ações e, principalmente, pessoas por quem tenho elevada estima e consideração.

Não seria justo, por exemplo, fazer agradecimentos a pessoas sem citar os meus pais, porque deles veio a base para a construção dos conhecimentos que culminaram com este livro.

Ao Exército Brasileiro, manifesto meus mais profundos agradecimentos pelas oportunidades de aprendizado, tanto em seus Estabelecimentos de Ensino quanto nas suas Organizações Militares onde tive a honra de servir. Servir em unidades de manutenção, ou em unidades ligadas à manutenção, me permitiu vivenciar e compreender a importância e os desafios desse importante ramo da profissão militar. Ao longo da minha carreira, essa nobre Instituição me disponibilizou excelentes condições de estudos e de aperfeiçoamentos, que contribuíram para a minha compreensão sobre os fundamentos técnicos e científicos do conteúdo apresentado neste livro. Desse modo, foi possível compartilhar tal metodologia de uma maneira estruturada para um público-alvo de tão alto nível acadêmico, como o são os alunos do Curso de Graduação em Engenharia Mecânica da Universidade das Forças Armadas – ESPE. Para mim, foi uma honra poder dirigir-me a tão seletivo grupo de estudantes e de companheiros docentes, durante o curso de Engenharia em Manutenção.

A todos os meus professores e instrutores pela escolha de suas nobres profissões, cujos esforços foram fundamentais para que seus alunos pudessem absorver o melhor disponível em cada ramo do conhecimento escolar, acadêmico e profissional. Muito obrigado! São todos para mim importantes referências.

Aos meus amigos e aos meus familiares, pela compreensão de que a minha ausência a muitos momentos importantes do seu sagrado convívio ocorreu porque eu estava em processo de construção de conhecimentos e de aprimoramento profissional. Obrigado a todos pelo incentivo!

Um especial agradecimento ao amigo Cel Alexandre de Oliveira Bleasby, que desde sua época de Cadete compartilha comigo seus conhecimentos sobre a lide castrense, me permitindo uma adequada compreensão sobre a profissão militar. Meu “muito obrigado”, extensivo à sua família.

Durante o Curso de Graduação, tive uma ajuda fundamental dos amigos e companheiros de formação na AMAN: Tenente-Coronel Marcos Paulo, Major Terra, Major Vinícius, Major Elson, Major Charles e Major Wander, todos do Exército Brasileiro. Meus amigos, muito obrigado por tudo.

Al Ejército Ecuatoriano, por recibirme en su tan prestigiosa Universidad de las Fuerzas Armadas – ESPE, permitiéndome contribuir con la formación de sus talentos humanos y poner un pequeño ladrillo en el grande y fuerte edificio de la ciencia y de la tecnología de su hermoso país.

Al Señor Lcdo. Xavier Chinga Mármol, quien con su inmensa ayuda me permitió aprender los muchos detalles del idioma español. Desde el inicio del proceso, el Señor Xavier Chinga, con elevado profesionalismo, contribuyó para que esta obra pudiera presentarse sin defectos del idioma para que su contenido sea expuesto de una manera clara para los usuarios y lectores. ¡Muchas gracias!

A la Comisión Editorial de la UFA-ESPE, integrada por: Tcrn. IGEO. Patricio Xavier Molina Simbaña, Ph. D.; Tcrn. de EM. Henry Omar Cruz Carrillo, Ph. D.; Ing. Sandra Patricia Galarza Torres, Ph. D.; Lcdo. Klever Antonio Bravo Calle, Dr.; Lcda. Aída Noemí Bedón Bedón, Ph. D.; Ing. David Vinicio Carrera Villacrés, Ph. D.; Ing. Lorenzo Adalid Armijos Robles, Mgs.; Lcdo. Carlos María Noceda Alonso, Ph. D.; Ing. Héctor Cochise Terán Herrera, Mgs.; Ing. Marco Javier Flores Calero, Ph. D.; Ing. Sonia Elizabeth Cárdenas Delgado, Ph. D; y al Ing. Milton Vinicio Uday Patiño, Mgs., por la aprobación de la obra y su inclusión en el Repositorio de la Universidad, de esta manera el contenido estará disponible para quien desee utilizar este libro como herramienta de conocimiento.

Al Señor Dr. David Andrade, por la sugerencia en escribir este libro y por el curso de redacción científica en el Instituto Tecnológico Superior Yavirac.

Por fim, a todos que embora não citados explicitamente nesta seção de agradecimentos, me ajudaram na construção e na organização metodológica de conhecimentos, muito obrigado! ¡Muchas gracias a todos!

Mantenimiento basado en confiabilidad

La expresión “Mantenimiento Basado en Confiabilidad (MBC)” es una adaptación de su correspondiente en inglés *Reliability Centered Maintenance (RCM)*, que es una técnica de mantenimiento en la cual se emplean métodos estadísticos para la determinación de la vida útil de un sistema, subsistema o de un componente. La confiabilidad está relacionada con las leyes de distribución de fallas y permite estimar la probabilidad de que un sistema se mantenga en funcionamiento más allá de un tiempo especificado.

Este libro busca presentar al lector una breve explicación respecto las aplicaciones estadísticas en el control de mantenimiento de una gran variedad de equipamientos, sean ellos industriales, de transporte o cualquier tipo de sistema sometido al mantenimiento correctivo, preventivo, predictivo o modificador. También, pretende ilustrar algunos métodos de cálculo de disponibilidad de equipamientos, de control de stock de suministros de mantenimiento y finalmente, hace una breve introducción respecto a las potenciales aplicaciones de aprendizaje de máquina que pueden ser adoptadas por la Industria 4.0.

El libro dispone al público interesado algunos conocimientos matemáticos y referentes a procesos estocásticos, complementados por códigos computacionales en lenguaje Python, para que el lector pueda automatizar sus análisis y desarrollar sus propias soluciones de mantenimiento. Además, se presenta una amplia bibliografía de autores e instituciones que se dedican al mantenimiento basado en confiabilidad, donde se puede profundizar el conocimiento mediante ejemplos prácticos que deben ser consultados siempre que surja la necesidad de apoyarse con un mayor soporte teórico al respecto.

Alexandre Galo Lopes

ISBN: 978-9942-765-61-1



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA