



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MECÁNICA

PREDICCIÓN DEL TIEMPO DE VIDA REMANENTE DE ESTRUCTURAS SUJETAS A FATIGA EN BASE AL MONITOREO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL MECANICO

SEBASTIÁN MIGUEL GARCÍA CORNEJO

PROFESOR GUÍA:
ENRIQUE LÓPEZ DROGUETT

MIEMBROS DE LA COMISIÓN:
VIVIANA MERUANE NARANJO
ALEJANDRO ORTIZ BERNARDIN

SANTIAGO DE CHILE
2017

PREDICCIÓN DEL TIEMPO DE VIDA REMANENTE DE ESTRUCTURAS SUJETAS A FATIGA EN BASE AL MONITOREO

Durante la vida de un componente, cambios micro estructurales emergen y evolucionan en el tiempo. Actualmente, se utilizan modelos clásicos de degradación empíricos que permiten monitorear el daño del componente. Sin embargo, estos utilizan indicadores de daño que no pueden ser detectados en etapas tempranas de degradación y cuando finalmente pueden ser detectadas la vida del componente esta próxima a acabar. Dado lo anterior, es de gran interés detectar el daño lo antes posible. Con la implementación de un modelo de redes bayesianas dinámicas es posible diagnosticar el daño de una estructura observando la evolución de los precursores de daño.

En el presente trabajo se busca implementar un modelo para el diagnóstico y pronóstico del daño de una bomba sujeta a fatiga en base a distintas señales recogidas de la bomba: vibraciones, caudal, potencia, intensidad y presión diferencial. El modelo mencionado fue construido en base a redes bayesianas dinámicas que son capaces de analizar el comportamiento no lineal y no gaussiano de la degradación por fatiga y también capaces de mezclar distintos tipos de evidencia (señales) para reducir la incertidumbre acerca del daño de la estructura.

El proceso de inferencia en la red bayesiana dinámica es modelado a través del uso de la técnica de filtro de partículas utilizando el software R para la implementación del modelo. Finalmente, con los resultados obtenidos es posible realizar la predicción del tiempo de vida remanente de la estructura sujeta a fatiga.

El presente informe da cuenta del trabajo paso a paso realizado por el alumno para llegar al resultado final. Se realiza en primer lugar una presentación de los antecedentes teóricos que son utilizados en la realización del trabajo, junto a la metodología desarrollada.

Se da paso luego a la presentación de los resultados obtenidos por el alumno, los cuales pueden ser divididos en dos etapas, una primera etapa de recepción y tratamiento de los datos duros para encontrar información útil para desarrollar la segunda etapa, donde se lleva a cabo el desarrollo del filtro de partículas para obtener finalmente un pronóstico de la vida de la bomba estudiada y la determinación de la vida útil remanente de esta.

Los resultados obtenidos son considerados positivos, ya que se logra una buena estimación del deterioro y tiempo de falla de la bomba y se cumplen los objetivos del trabajo de título.

Agradecimientos.

A mis padres, quienes me dejaron estudiar la carrera que yo quisiera elegir y me dieron el apoyo para estudiarla sin preocupaciones adicionales. Por apoyarme y escucharme cuando tuve dudas y permitirme realizar un intercambio.

A mi profesor guía, Enrique López Droguett, por permitirme realizar este trabajo de título y tener la mejor disposición frente a las dudas que surgieron en el camino.

Por último, a mis amigos y familia que me acompañaron en esta etapa de mi vida.

Tabla de Contenido

1. Introducción.....	1
Objetivos:.....	1
Objetivo General.....	1
Objetivos Específicos.....	1
Alcances:.....	2
2. Antecedentes y discusión bibliográfica.....	2
2.1. Redes Bayesianas.....	2
2.1.1. Razonamiento con Redes Bayesianas.....	3
2.1.2. Representación de la distribución de probabilidad conjunta.....	4
2.1.3. Independencias condicionales.....	5
2.1.4. Inferencia de una red bayesiana.....	6
2.2. Redes Bayesianas Dinámicas.....	7
2.3. Filtro de partículas.....	9
2.4. Parámetros Estadísticos.....	11
2.5. Empirical Mode Decomposition (EMD).....	12
2.6. Support Vector Machines Regression (SVR).....	14
3. Montaje experimental.....	18
4. Metodología.....	20
5. Resultados obtenidos.....	23
5.1. Distancia de mahalanobis en base a vibraciones.....	23
5.2. Distancia de Mahalanobis en base a la eficiencia.....	28
5.3. Empirical Mode Decomposition (EMD).....	29
5.4. Implementación del modelo.....	31
5.4.1. Modelo del sistema (System Model).....	31
5.4.2. Modelo de medición (Measurement model).....	33
5.5. Diagnóstico y pronóstico de la degradación.....	33
5.6. Determinación de la vida remanente de la bomba.....	34
6. Discusión y conclusiones.....	36
7. Bibliografía.....	37
Anexos.....	I
Anexo A.....	II
Anexo B. Códigos.....	V

MATLAB: Obtencion de RMS bandas.	V
MATLAB: MD bandas RMS.	VI
MATLAB: Parámetros eficiencia.	VII
MATLAB: MD Eficiencia y EMD.	VIII
R: Filtro Partículas.	IX

Índice de Ilustraciones.

Figura 2.1: Ejemplo red bayesiana.	2
Figura 2.2: Tipos de razonamiento, traducido de [4].	4
Figura 2.3: Cadena causal.....	5
Figura 2.4: Causa común.....	5
Figura 2.5: Efecto común.	6
Figura 2.6: Cadena de 3 nodos.	7
Figura 2.7: Ejemplo DBN.....	8
Figura 2.8: Representación de una DBN con evidencia de múltiples variables [1].	9
Figura 2.9: Ejemplo EMD.	14
Figura 2.10: Función de pérdida ε -insensitive [12].....	16
Figura 3.1: Cámara de prueba.	18
Figura 3.2: Configuración de los acelerómetros.....	18
Figura 3.3: Sensores utilizados.....	19
Figura 4.1: Metodología.	20
Figura 4.2: Modelo DBN.....	21
Figura 4.3: Cálculo de tiempo de vida remanente [13].	22
Figura 5.1: Análisis espectral.	24
Figura 5.2: Análisis espectral 0 - 500 [Hz].....	25
Figura 5.3: Representación bandas.....	26
Figura 5.4: Distancia de mahalanobis en base a los datos de vibración.	27
Figura 5.5: Logaritmo de la distancia de mahalanobis.....	27
Figura 5.6: Parámetros eficiencia.	28
Figura 5.7: Distancia de mahalanobis en base a la eficiencia de la bomba.....	29
Figura 5.8: Tendencia deficiencia.	30
Figura 5.9: Tendencia vibraciones.....	31
Figura 5.10: Regresión polinomial.	32
Figura 5.11: Resultados SVM.	33
Figura 5.12: Diagnóstico y pronóstico.	34
Figura 5.13: Tiempo de vida remanente.....	35

1. Introducción.

Durante la vida de un componente, cambios micro estructurales emergen y evolucionan en el tiempo. Estos cambios llevan al desgaste y degradación de un componente, lo que puede ocasionar una falla del sistema al cual pertenece el componente, la cual puede ser incluso catastrófica. Esto hace que sea importante monitorear el estado de degradación de los componentes en algunos sistemas. Actualmente, para realizar este monitoreo se utilizan modelos clásicos de degradación empíricos. El problema de estos modelos es que requieren indicadores de daño directos, como la evolución del tamaño de grieta para realizar el monitoreo del componente. Estos indicadores son marcas de daño observables, las cuales suelen ser detectadas una vez que el componente ya se encuentra en un estado de degradación importante y se encuentra próximo a la ocurrencia de una falla.

Debido a esto, es de gran interés tratar de diagnosticar el daño lo antes posible. Se han estudiado distintos métodos para realizar este diagnóstico, uno de los cuales a través del monitoreo de indicadores de daño indirectos o precursores de daño [1]. Este estudio busca desarrollar un marco de referencia para realizar el diagnóstico y pronóstico del estado de degradación de un componente, en el cual se utiliza un modelo de redes bayesianas dinámicas (DBN) para modelar la evolución del sistema en el tiempo [2].

Es necesario ser capaz de relacionar las distintas variables que se observan como indicadores de daño indirecto con el estado interno del sistema, ya que se busca realizar el diagnóstico de daño sin la necesidad de utilizar indicadores directos. Este aspecto también es bien cubierto gracias a las DBN, ya que estas pueden correlacionar de buena manera la evolución dinámica del sistema con variables inciertas. Por otro lado, la estimación de la evolución del estado de degradación del componente puede ser realizada a través de la técnica de filtro de partículas [3].

De esta forma se puede modelar el sistema con ayuda de las DBN utilizando la técnica de filtro de partículas como mecanismo de inferencia de la red. Así, es posible diagnosticar el grado de degradación de un componente y la evolución del mismo para poder estimar el tiempo de vida remanente de la pieza para así programar el mantenimiento del sistema y evitar fallas catastróficas.

Objetivos:

Objetivo General.

Implementar un modelo para el diagnóstico y pronóstico del daño en estructuras sujetas a fatiga en base a señales de vibraciones y/o emisiones acústicas para predecir su tiempo de vida remanente.

Objetivos Específicos.

- Implementar el modelo de diagnóstico y pronóstico de la bomba en base a redes bayesianas dinámicas, utilizando el software R.

- Realizar el tratamiento de los datos brutos para obtener datos de tendencias necesarias para el modelo.
- Realizar el modelo de inferencia de la red mediante la técnica de filtro de partículas.
- Analizar un caso de estudio de una bomba sujeta al monitoreo de vibraciones.

Alcances:

Los alcances del trabajo abarcan la implementación del modelo para el análisis de un caso de estudio de una bomba centrífuga monitoreada en base a vibraciones, logrando diagnosticar su estado de degradación y predecir su tiempo de vida remanente.

2. Antecedentes y discusión bibliográfica.

2.1. Redes Bayesianas.

Las redes bayesianas o BN's (por sus siglas en inglés) son modelos gráficos útiles para el razonamiento bajo incertidumbre. Los nodos de la red bayesiana representan un set de variables aleatorias y sus arcos representan las dependencias condicionales entre variables.

Una red bayesiana debe ser un gráfico a cíclico, es decir, no debe ser posible volver a un mismo nodo a través de las conexiones entre variables.

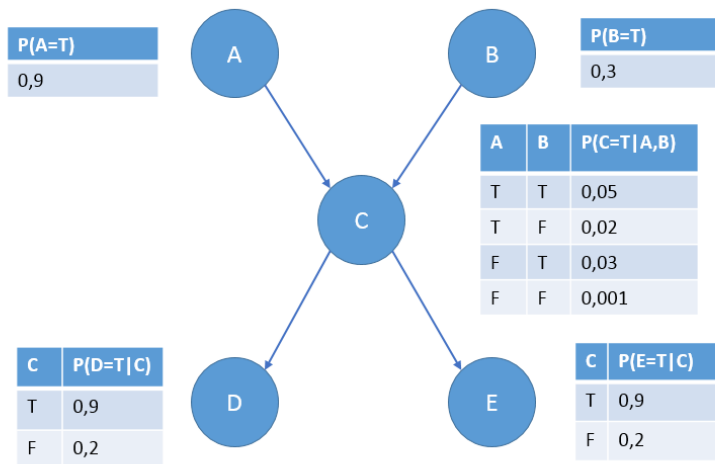


Figura 2.1: Ejemplo red bayesiana.

La Figura 2.1 muestra un ejemplo de una red bayesiana. La estructura de la red debe capturar relaciones cuantitativas entre las variables, dichas relaciones están representadas por los arcos que las unen, donde la dirección del arco indica la relación causa efecto. En este ejemplo se coincidirían solo variables discretas.

Los nodos de los cuales salen arcos hacia otro nodo son llamados padres del último, por ejemplo en la figura, los nodos A y B son padres de C y C es padre de los nodos D y E. A su vez, el nodo C es llamado hijo tanto del nodo A como del nodo B y los nodos D y E son hijos del nodo C. Cualquier nodo que sea padre es llamado nodo raíz. En este caso, los nodos A y B. Cualquier nodo sin hijos es llamado nodo hoja: D y E en este ejemplo.

Luego de definir las relaciones entre los nodos de la red, es necesario cuantificar las relaciones entre los mismos. Para esto, se deben especificar las distribuciones de probabilidad condicional para cada nodo. En nuestro ejemplo, dado que se trabaja con variables discretas, esto se traduce en tablas de probabilidad condicional (CPT por sus siglas en inglés).

Una propiedad importante que se asume generalmente en una red bayesiana es la propiedad markoviana, lo que se traduce en que no existen dependencias directas en la red si no están explicitadas por medio de los arcos. Las redes bayesianas con esta propiedad también reciben el nombre de Independence-maps (I-maps).

2.1.1. Razonamiento con Redes Bayesianas.

Cuando recibimos información sobre alguna variable, nos gustaría actualizar nuestra creencia sobre el resto de las variables. Este procedimiento se hace mediante un “flujo de información” a través de la red.

Existen distintos tipos de razonamiento según el flujo de información. Por ejemplo, el diagnóstico es un razonamiento que se estructura partiendo desde los síntomas hacia las causas. Por otro lado, un razonamiento predictivo utilizaría una nueva información sobre las causas para actualizar las creencias sobre los síntomas.

La Figura 2.2 muestra diferentes tipos de razonamiento.

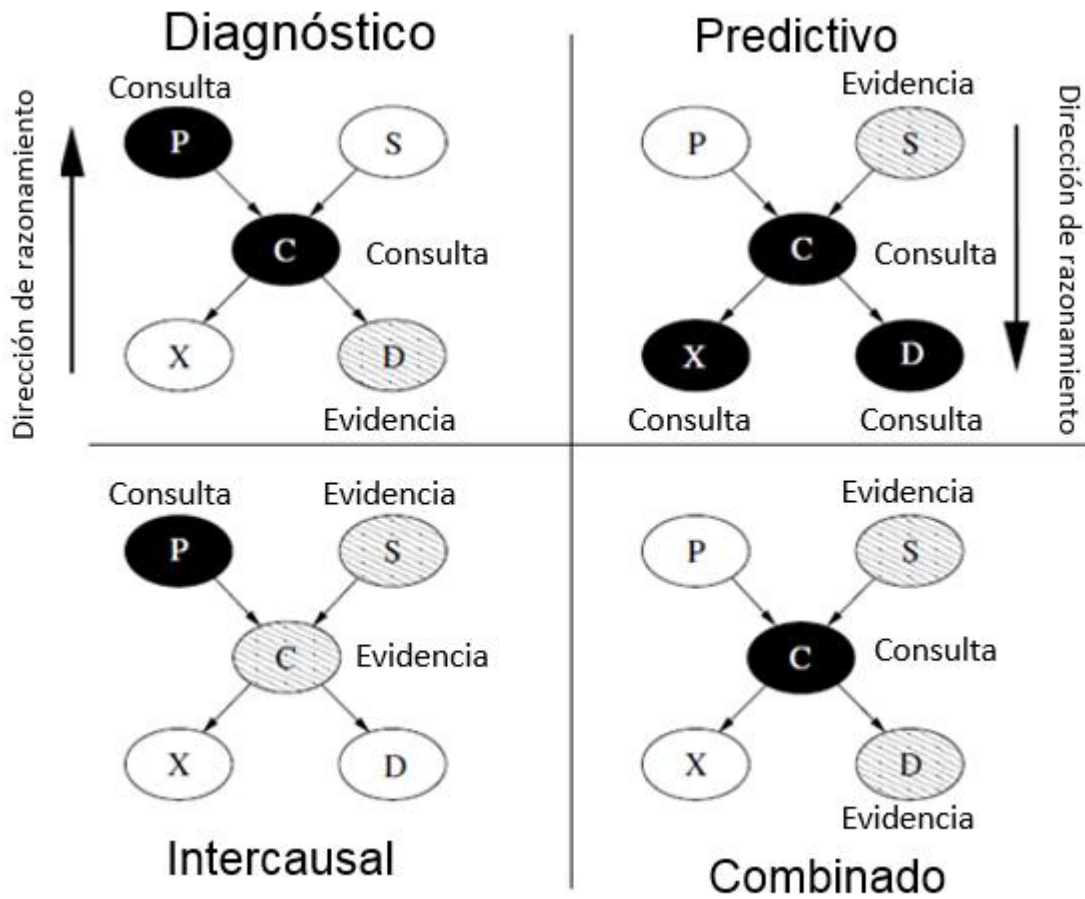


Figura 2.2: Tipos de razonamiento, traducido de [4].

2.1.2. Representación de la distribución de probabilidad conjunta.

Si consideramos una red bayesiana compuesta por n nodos, X_1 a X_n en tal orden, un valor particular en la distribución conjunta estará representado por $P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = P(x_1, x_2, \dots, x_n)$.

La regla de la cadena de la teoría de probabilidades nos permite factorizar la distribución conjunta según

$$\begin{aligned}
 P(x_1, x_2, \dots, x_n) &= P(x_1) * P(x_2|x_1) \dots * P(x_n|x_1, \dots, x_{n-1}) \\
 &= \prod_i P(x_i|x_1, \dots, x_{i-1})
 \end{aligned}$$

Luego, como la estructura de la red implica que el valor de un nodo sólo está condicionado a los valores de sus padres, la ecuación anterior se reduce a:

$$P(x_1, x_2, \dots, x_n) = \prod_i P(x_i | \text{Padres}(X_i))$$

2.1.3. Independencias condicionales.

Las redes bayesianas que satisfacen la propiedad de markov (I-Maps) expresan independencias condicionales explícitas en las distribuciones de probabilidad. Estas independencias pueden ser aprovechadas en los casos de cadena causal, causa común y efecto común.

Cadena causal.

Consideremos una cadena causal como se muestra en la Figura 2.3.

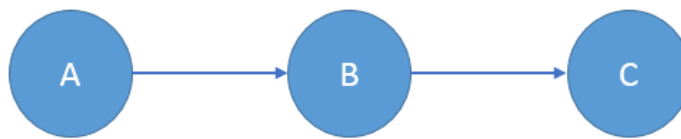


Figura 2.3: Cadena causal.

Esta cadena da paso a una independencia condicional ya que, debido a la propiedad de markov se cumple que:

$$P(C|A \wedge B) = P(C|B)$$

Es decir, dado B, no nos interesa saber A en nuestra creencia sobre C.

Causa común.

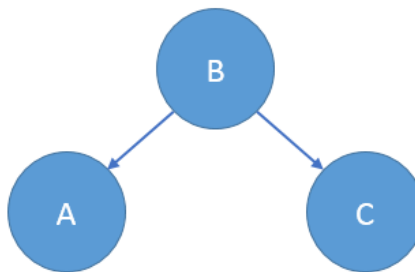


Figura 2.4: Causa común.

La Figura 2.4 representa una independencia común de causa común, en la cual B es la causa tanto de A como de C. Este caso da la misma independencia condicional del caso anterior:

$$P(C|A \wedge B) = P(C|B)$$

Efecto común.

Por último, en el caso de efecto común tendremos:

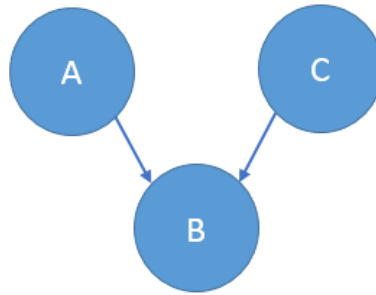


Figura 2.5: Efecto común.

En este caso, la independencia condicional es contraria a los casos anteriores:

$$P(A|C \wedge B) \neq P(A|C)$$

Entonces, si B ocurre, entonces si A no ocurre, esto aumentaría las probabilidades de C ocurra.

2.1.4. Inferencia de una red bayesiana.

Inferencia exacta en cadenas.

Para tener una visión de cómo funciona la inferencia en una red bayesiana se estudiará primero la inferencia exacta en cadenas de una red.

El primer caso a estudiar es el caso más simple, con una cadena de dos nodos $A \rightarrow B$.

En caso de realizar la inferencia desde el padre al hijo, la actualización de nuestra creencia sobre B $Bel(B)$ puede obtenerse directamente de la CPT.

Por otro lado, en caso contrario, donde se realiza la inferencia desde el hijo al padre, para actualizar nuestra creencia sobre A, $Bel(A)$ es necesario aplicar el teorema de Bayes:

$$\begin{aligned} Bel(A = a) &= P(A = a|B = b) \\ &= \frac{P(B = b|A = a) * P(A = a)}{P(B = b)} \end{aligned}$$

Según el teorema, $P(A = a)$ es la información a priori sobre A y $P(B = b|A = a)$ es la verosimilitud. El término $P(B = b)$ es la constante de normalización.

En el caso de una cadena de tres nodos como el ilustrado en la Figura 2.6, si tenemos evidencia sobre el nodo A y queremos actualizar nuestra creencia sobre el nodo C al movernos en la misma dirección de los arcos sólo debemos aplicar la regla de la cadena:

$$Bel(C) = P(C|A = a) = \sum_{B=b} P(C|B) * P(B|A = a)$$

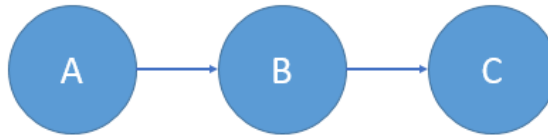


Figura 2.6: Cadena de 3 nodos.

Por otro lado, en caso de tener información sobre el nodo hoja C, la actualización de nuestra creencia sobre A se realiza aplicando el teorema de Bayes junto a la regla de la cadena:

$$\begin{aligned}
 Bel(A = a) &= P(A = a|C = c) \\
 &= \frac{P(C = c|A = a) * P(A = a)}{P(C = c)} \\
 &= \frac{\sum_{Y=y} P(C = c|B = b, A = a) * P(B = b|A = a) * P(A = a)}{P(C = c)} \\
 &= \frac{\sum_{Y=y} P(C = c|B = b) * P(B = b|A = a) * P(A = a)}{P(C = c)}
 \end{aligned}$$

Con los tipos de inferencia exacta ya mencionados es posible tener una noción de cómo funciona la inferencia en las redes bayesianas.

Inferencia aproximada con simulación estocástica.

La simulación estocástica utiliza la red para generar un gran número de casos a partir de la distribución de la red. La probabilidad posterior de algún nodo específico es estimada utilizando dichos casos. Acorde a la ley de los grandes números, mientras más casos sean generados, mayor exactitud tendrá la probabilidad estimada.

El algoritmo más simple de muestreo es el algoritmo de muestreo lógico o logic simple (LS).

Este algoritmo genera casos seleccionando de forma aleatoria los valores para cada nodo, dándoles peso según la probabilidad de que el nodo tome dicho valor. El algoritmo recorre los nodos desde la raíz hasta las hojas, por lo tanto en cada paso, la probabilidad con peso asociado será la priori o el valor de la CTP para los valores de los padres elegidos.

Una vez que todos los nodos son visitados se origina un caso. Para estimar la probabilidad de $P(X|E)$ con un valor de muestreo $P'(X|E)$ donde E representa la evidencia, se calcula la razón entre el número de casos donde tanto X como E son verdaderos versus el número de casos donde sólo E es verdadero.

El problema principal de este algoritmo es que cuando la evidencia E es muy poco probable la mayoría de los casos deben ser descartados ya que estos no contribuyen en el conteo de casos.

2.2. Redes Bayesianas Dinámicas.

Las redes bayesianas dinámicas (DBN) son una forma de extender una red bayesiana para modelar distribuciones de probabilidad con variables semi-infinitas, es decir, permiten modelar sistemas dinámicos. Éstas permiten relacionar variables entre sí a través de pasos de tiempo. La Figura 2.7 muestra una DBN simple. En esta, los nodos de color oscuro representan las variables observables, como mediciones del sistema, mientras que los nodos claros representan las variables del estado de un sistema no observables.

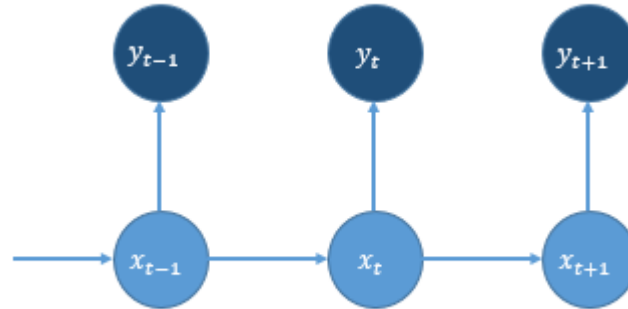


Figura 2.7: Ejemplo DBN.

Una DBN nos permite obtener la función de densidad de probabilidad (pdf) posterior de forma recursiva en dos etapas, predicción y actualización. Para esto, es necesario conocer las relaciones entre los nodos. El modelo que relaciona al sistema con su estado anterior recibe el nombre de system model, mientras que el modelo que relaciona el estado interno con la variable observable recibe el nombre de measurement model.

Supongamos que conocemos la pdf $p(x_{t-1}|y_{1:t-1})$ correspondiente al instante de tiempo $t - 1$. La etapa de predicción involucra utilizar el modelo del sistema (system model) para obtener la pdf a priori del estado en el tiempo t utilizando la ecuación de Chapman-Kolmogorov:

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1}) * p(x_{t-1}|y_{1:t-1}) * dx_{t-1} \quad (9)$$

En la ecuación anterior se aprovecha que $p(x_t|x_{t-1}, y_{1:t-1}) = p(x_t|x_{t-1})$ dado que el modelo del sistema describe un proceso de Markov de primer orden.

En la etapa de actualización, una nueva medición y_t se obtiene en el instante de tiempo t , lo que nos permite actualizar nuestra a priori a través del teorema de Bayes:

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t) * p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \quad (10)$$

Donde la constante de normalización

$$p(y_t|y_{1:t-1}) = \int p(y_t|x_t) * p(x_t|y_{1:t-1}) dx_t \quad (11)$$

Depende de la función de verosimilitud $p(y_t|x_t)$ dado por el measurement model.

Cuando contamos con evidencias de múltiples variables, como en el caso de la Figura 2.8, se requiere un measurement model integrado, en este caso $P(E_k, AE_k | D_k^*)$. Según la regla de la cadena, podemos expandir la probabilidad según:

$$P(E_k, AE_k | D_k^*) = P(E_k | AE_k, D_k^*) * P(AE_k | D_k^*) * P(D_k^*) \quad (12)$$

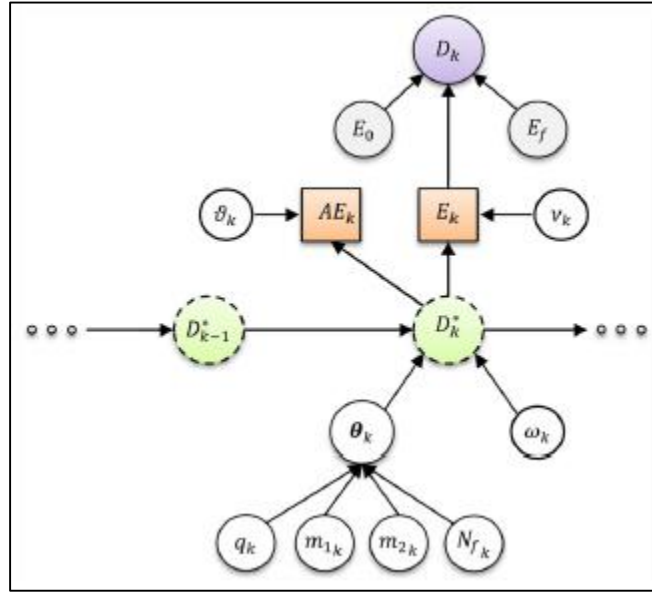


Figura 2.8: Representación de una DBN con evidencia de múltiples variables [1].

Debido a la naturaleza de las variables, en algunos casos es razonable creer que estas son independientes entre sí, por lo que el measurement model integrado termina siendo el siguiente:

$$P(E_k, AE_k | D_k^*) \propto P(D_k | D_k^*) * P(AE_k | D_k^*) * P(D_k^*) \quad (13)$$

De esta forma es posible integrar la información disponible de todas las variables para obtener la estimación del estado actual.

La solución de estas ecuaciones no puede ser obtenida de forma analítica, por lo que se han desarrollado una serie de algoritmos para obtener una aproximación.

El método de filtro de partículas permite una aproximación óptima a la solución bayesiana. Este método será explicado a continuación.

2.3. Filtro de partículas.

El filtro de partículas es un método de Montecarlo Secuencial bastante útil para estudiar la evolución de fenómenos no lineales no gaussianos.

La idea de los filtros de partículas es aproximar nuestra creencia de cierta variable a estudiar a través de la generación de una muestra de partículas con pesos asociados según su probabilidad de ocurrencia.

En este sentido, cada partícula generada es un posible estado de la variable estudiada en un instante de tiempo.

La pdf posterior es aproximada entonces según:

$$p(x_t|y_{1:t}) \approx \sum_{i=1}^{N_s} \omega_t^i \delta(x_t, x_t^i) \quad (14)$$

Con X_t^i la i -ésima muestra de X_t generada.

Los pesos ω_k^i son elegidos según el principio de importance sampling. En importance sampling asumimos que es difícil obtener muestras de la distribución objetivo $\pi(x)$. Se asume que $P(x) \propto \pi(x)$. Entonces en lugar de realizar un sampling desde la distribución objetivo, las muestras son generadas desde una distribución de importancia $q(x)$ propuesta.

Así, los pesos se obtienen según:

$$\omega_t^i \propto \frac{p(x_{1:t}^i|y_{1:t})}{q(x_{1:t}^i|y_{1:t})} \quad (15)$$

El término $P(x_{1:t}^i|y_{1:t})$ puede ser obtenido de forma recursiva utilizando el teorema de Bayes. Nos interesa además, que la distribución propuesta $q(x)$ pueda ser obtenida de la misma forma, lo que nos da:

$$\begin{aligned} \omega_t^i &\propto \frac{P(y_t|x_t^i) * P(x_t^i|x_{t-1}^i) * P(x_{1:t-1}^i|y_{1:t-1})}{q(x_t^i|x_{1:t-1}^i, y_{1:t}) * q(x_{1:t-1}^i|y_{1:t-1})} \\ &= \frac{P(y_t|x_t^i) * P(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{1:t-1}^i, y_{1:t})} * \omega_{t-1}^i \end{aligned} \quad (16)$$

En el caso de filtrado, no es necesario guardar toda la información pasada, sino sólo el último dato. Con esto se tiene que $q(x_t^i|x_{1:t-1}^i, y_{1:t}) = q(x_t^i|x_{t-1}^i, y_t)$, con lo cual sólo es necesario guardar x_{t-1}^i , simplificando los pesos en:

$$\omega_t^i = \frac{P(y_t|x_t^i) * P(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, y_t)} * \omega_{t-1}^i \quad (17)$$

Este método, en el cual se aplica el importance sampling es conocido como sequential importance sampling (SIS). Un problema asociado a la técnica de filtro de partículas por SIS es conocido como el problema de degeneración, en el cual luego de unas pocas iteraciones, todas las partículas a excepción de una tendrán un peso despreciable, lo que se traduce en que hay un gran esfuerzo computacional en actualizar partículas que no tendrán ningún efecto en la aproximación de $P(x_t|y_{1:t})$.

Un método alternativo para evitar el problema de regeneración es utilizar el método de resampling (SIR). La idea de este método es eliminar las partículas que tienen pesos pequeños y concentrarse en las partículas de mayor peso.

Este método genera una nueva muestra de partículas a través de muestreo con reemplazo basado en el peso de las partículas, generando una muestra de tamaño N_s tal que $\Pr(x_t^{i*} = x_t^j) = \omega_t^j$.

La nueva muestra es i.d.d., por lo cual los nuevos pesos son reseteados como $\omega_t^i = 1/N_s$. Luego se procede según el método anterior.

El algoritmo de filtro de partículas SIR sigue entonces el siguiente mecanismo:

1. Se realiza un muestreo de N partículas x^i de la distribución de importancia elegida: $x_t^i \sim q(x_t | x_{t-1}, y_t)$, donde cada una de las partículas de la muestra es una hipótesis sobre el estado del sistema.
2. Se realiza la actualización de cada partícula según $p(x_t^i | x_{k-1}^i)$
3. Se obtiene el peso de cada partícula según la ecuación (17)
4. Se realiza el proceso de resampling.
5. Se repite desde el paso 2.

Si bien este método soluciona el problema de degeneración, también introduce otros problemas prácticos. Por ejemplo, como el muestreo del resampling se realiza según los pesos de las partículas, las partículas de mayor peso serán seleccionadas dentro de la muestra más veces, lo que implica una pérdida de diversidad de las partículas, ya que la muestra contendrá partículas que se repiten varias veces.

2.4. Parámetros Estadísticos.

A continuación se detallan los parámetros estadísticos utilizados en este trabajo, los cuales son reconocidas ampliamente en el ambiente científico. Las definiciones fueron extraídas de [6] y [7].

- Valor medio (Mean value).
Promedio de la señal.

$$S_{mean} = \frac{1}{N} \sum_{i=1}^N S_i \quad (18)$$

- Valor pico-a-pico (Peak to peak value).
Distancia entre la amplitud máxima y la amplitud mínima.

$$S_{pico-pico} = S_{max} - S_{min} \quad (19)$$

- Media Cuadrática (Root Mean Square (RMS)).
Describe la energía contenida en la señal. Se utiliza para evaluar la condición general de un componente.

$$s_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N (s_i)^2} \quad (20)$$

- Desviación estándar (Standard Deviation).
Es una medida de la dispersión de los valores de la señal.

$$s_{std} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N |s_i - s_{mean}|^2} \quad (21)$$

- Factor de cresta (Crest factor).
Razón entre el valor del peak de una señal y el RMS de la misma.

$$CF = \frac{s_{peak}}{s_{rms}} \quad (22)$$

- Distancia de Mahalanobis.

Medida de distancia que toma en consideración la correlación existente entre las variables aleatorias que componen un vector para determinar la similitud de este en comparación a una línea base.

La distancia de Mahalanobis entre dos vectores \vec{x}, \vec{y} se define como [8]:

$$D_M(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})} \quad (23)$$

Donde:

S = Matriz de covarianza.

2.5. Empirical Mode Decomposition (EMD).

El empirical mode decomposition es un metodo que logra descomponer una señal en el dominio temporal. Este método es útil para el análisis de señales no lineales y no estacionarias [9].

El método permite descomponer la señal en una serie de funciones llamadas Intrinsic Mode Functions (IMFs) sin abandonar el dominio temporal, logrando descomponer la señal en modos de alta frecuencia, que vienen asociadas con el ruido de la señal, hasta modos de baja frecuencia, entregando un resto que da cuenta de la tendencia básica de la señal.

La señal original estaría compuesta por N IMFs según [10]:

$$x(t) = \sum_i^N c_i(t) + r(t)$$

Una IMF es definida como una función que satisface dos condiciones [11]:

- En el total de datos, el número de extremos y el número de veces que la función cruza el valor 0 deben ser igual o diferenciarse en 1.
- En cualquier punto, el valor promedio de la envolvente definida por el máximo local y la envolvente definida por el mínimo local debe ser 0.

En la figura 2.9 es posible observar un ejemplo de la descomposición de la señal, donde se aprecia como el resto de esta es capaz de capturar la tendencia general de la señal.

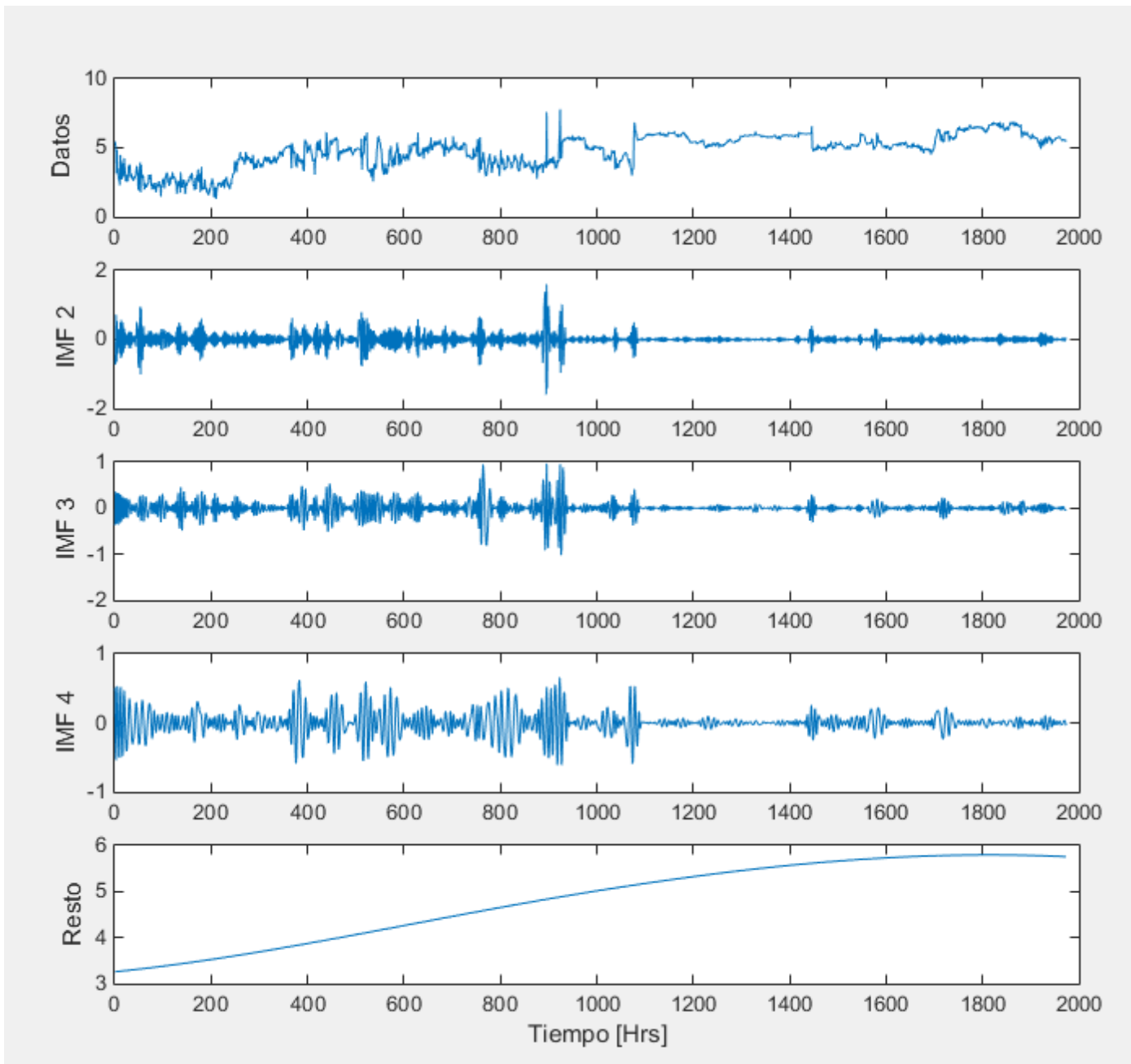


Figura 2.9: Ejemplo EMD.

2.6. Support Vector Machines Regression (SVR).

El desarrollo de esta sección fue sustraída de la referencia [12].

Los support vector machine regression (SVR) son métodos de aprendizaje de maquina supervisado, los cuales realizan una función de aproximación en base a conceptos de Teoría de aprendizaje estadístico junto a un conjunto de datos de entrenamiento disponibles del caso que se desea estudiar.

El objetivo de los SVR es encontrar una función de mapeo $y = f(x)$ que permita relacionar de buena forma un vector de entrada x con un valor de salida y . La información disponible para encontrar dicha función es proporcionada por un set D , que contiene sets de vectores de entrada x con su correspondiente valor de salida y .

El set D es utilizado en un proceso de aprendizaje para encontrar la función de mapeo $y = f(x)$. En el caso de las técnicas de SVR, esto se traduce en un problema de optimización convexo.

El primer caso a estudiar es el caso de una separación lineal. En este caso, es necesario encontrar el hiperplano lineal de la regresión que mejor se ajuste a los datos disponibles (set de entrenamiento D), para que pueda ser utilizado luego en la predicción al obtener nueva evidencia. La función del hiperplano puede ser expresada como:

$$f(x) = \mathbf{w}^T \mathbf{x} + b \tag{25}$$

Donde:

\mathbf{w} = factor de ponderación o peso.

b = punto de intercepción del modelo.

El error empírico del entrenamiento viene dado por la función de pérdida ε -insensitive:

$$e(x, y, f) = \max(0, |y - f(x)| - \varepsilon) \tag{26}$$

La fórmula anterior ignora errores menores a ε para $y - f(x)$, de esta forma, la función de pérdida ε -insensitive define un “tubo” de error nulo de radio ε y asigna un error igual a la distancia entre el punto y y el límite ε para puntos fuera de este.

Luego, es posible definir las variables *slack* ξ y ξ^* como:

$$\begin{cases} |y - f(x)| - \varepsilon = \xi, & \text{para puntos "sobre" el tubo.} \\ |y - f(x)| - \varepsilon = \xi^*, & \text{para puntos "bajo" el tubo.} \end{cases}$$

La figura 2.10 muestra la función de pérdida.

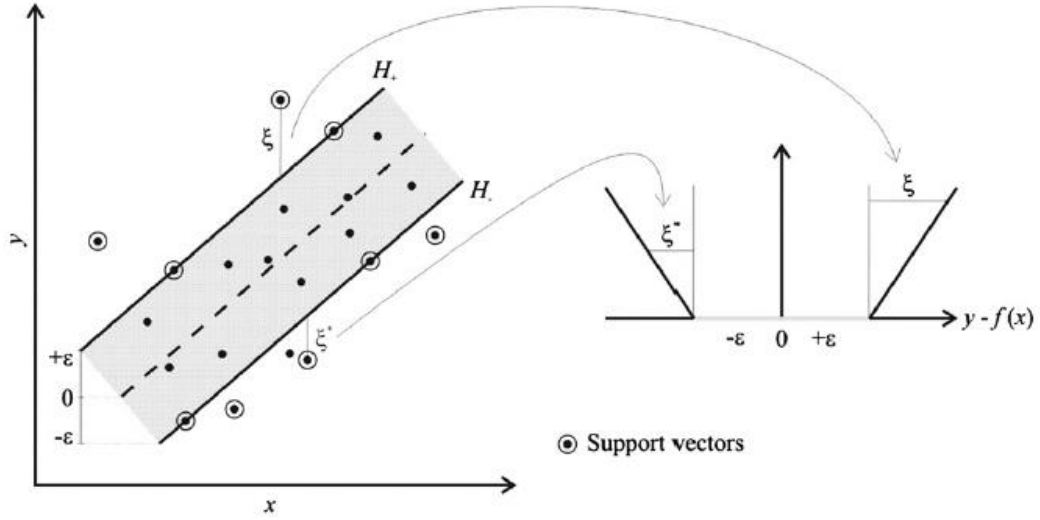


Figura 2.10: Función de pérdida ε -insensitive [12].

Luego, es necesario minimizar tanto el error empírico como la norma l_2 de los pesos ($\|\mathbf{w}\|^2$). El problema de optimización lineal cuadrático que representa la fase de entrenamiento es:

$$\text{Minimizar } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_t (\xi_t + \xi_t^*) \quad \mathbf{w}, b, \xi_t, \xi_t^* \quad (27)$$

sujeto a:

$$y_t - \mathbf{w}^T \mathbf{x}_t - b \leq \varepsilon + \xi_t \quad (28)$$

$$\mathbf{w}^T \mathbf{x}_t + b - y_t \leq \varepsilon + \xi_t^* \quad (29)$$

$$\xi_t \geq 0 \quad (30)$$

$$\xi_t^* \geq 0 \quad (31)$$

Donde C representa el trade-off entre los errores empíricos y de generalización.

Luego, el lagrangeano de la función primal es:

$$\begin{aligned} L_p(\mathbf{w}, b, \xi_t, \xi_t^*, \alpha, \alpha^*, \beta, \beta^*) & \quad (32) \\ &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_t (\xi_t + \xi_t^*) - \sum_t (\beta_t \xi_t + \beta_t^* \xi_t^*) \\ & - \sum_t \alpha_t (\mathbf{w}^T \mathbf{x}_t + b - y_t + \varepsilon + \xi_t) \\ & - \sum_t \alpha_t^* (y_t - \mathbf{w}^T \mathbf{x}_t - b + \varepsilon + \xi_t^*) \end{aligned}$$

Donde $\alpha_t, \alpha_t^*, \beta_t, \beta_t^*$ son los multiplicadores de lagrange asociados a las restricciones (28), (29), (30) y (31).

Es posible reducir la ecuación (32) introduciendo algunas igualdades proporcionadas por las condiciones complementarias KKT, con lo que se obtiene el lagrangeano dual L_D :

$$\begin{aligned} \text{Maximizar } L_D(\alpha, \alpha^*) & \quad (33) \\ &= -\frac{1}{2} \sum_t \sum_j (\alpha_t - \alpha_t^*)(\alpha_j - \alpha_j^*) \mathbf{x}_t^T \mathbf{x}_j - \sum_t (\varepsilon - y_t) \alpha_t \\ &\quad - \sum_t (\varepsilon + y_t) \alpha_t^* \end{aligned}$$

$$t, j = 1, 2, \dots, l$$

sujeeto a:

$$\sum_t (\alpha_t - \alpha_t^*) = 0$$

$$0 \leq \alpha_t \leq C$$

$$0 \leq \alpha_t^* \leq C$$

Finalmente, la solución del problema dual entrega la función de regresión estimada:

$$f_0(\mathbf{x}) = \mathbf{w}_0^T \mathbf{x} + b_0 = \sum_t (\alpha_{t0} - \alpha_{t0}^*) \mathbf{x}_t^T \mathbf{x} + b_0 \quad (34)$$

Donde el índice '0' da cuenta de la optimalidad.

Para el caso de un proceso de SVR no lineal, la idea es llevar los vectores de entrada $\mathbf{x}_t \in R^d$ a un *feature space* F de mayor dimensión a través de la función de mapeo $\Phi(\mathbf{x}_i)$ para luego resolver una regresión lineal en dicho espacio. En este caso, los productos punto $\mathbf{x}_t^T \mathbf{x}_j$ de la función (32) pueden ser reemplazados por $\Phi^T(\mathbf{x}_t) \Phi(\mathbf{x}_j)$. Además, las funciones producto punto son reemplazadas por funciones kernel $K(\mathbf{x}_t, \mathbf{x}_j)$ definidas en el espacio de entrada que implícitamente hace el mapeo de \mathbf{x} en F .

3. Montaje experimental.

Los datos proporcionados para el estudio de la bomba se originan de un experimento realizado en Estados Unidos por la Universidad de Maryland.

Con el objetivo de estudiar las dependencias entre bombas idénticas por ambiente común, proximidad y diseño idéntico se dispusieron 2 bombas centrífugas Jabsco 50840-0012 a bombear agua marina a temperaturas elevadas. El estudio se hizo de forma secuencial, se monitoreo una bomba hasta su falla para luego montar y monitorear la segunda bomba. De este modo, ambas fueron montadas en la misma cámara de prueba. La figura 3.1 muestra la cámara de prueba del experimento.



Figura 3.1: Cámara de prueba.

Las señales de vibración fueron recolectadas utilizando 3 acelerómetros de un eje, con una frecuencia de muestreo de 10240 [Hz]. La figura 3.2 muestra la configuración final de los acelerómetros en la bomba.

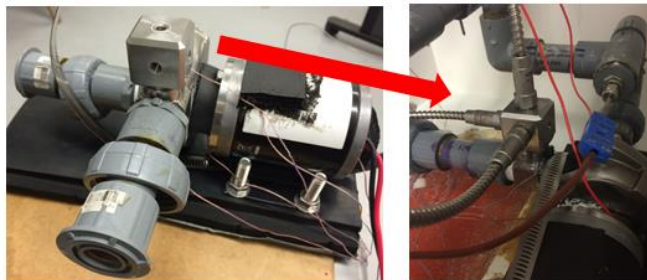


Figura 3.2: Configuración de los acelerómetros.

Además del monitoreo de las vibraciones, se monitorearon otras variables en el experimento como emisiones acústicas, temperatura y otras. Los distintos sensores utilizados en el experimento y su disposición se muestran en la figura 3.3.

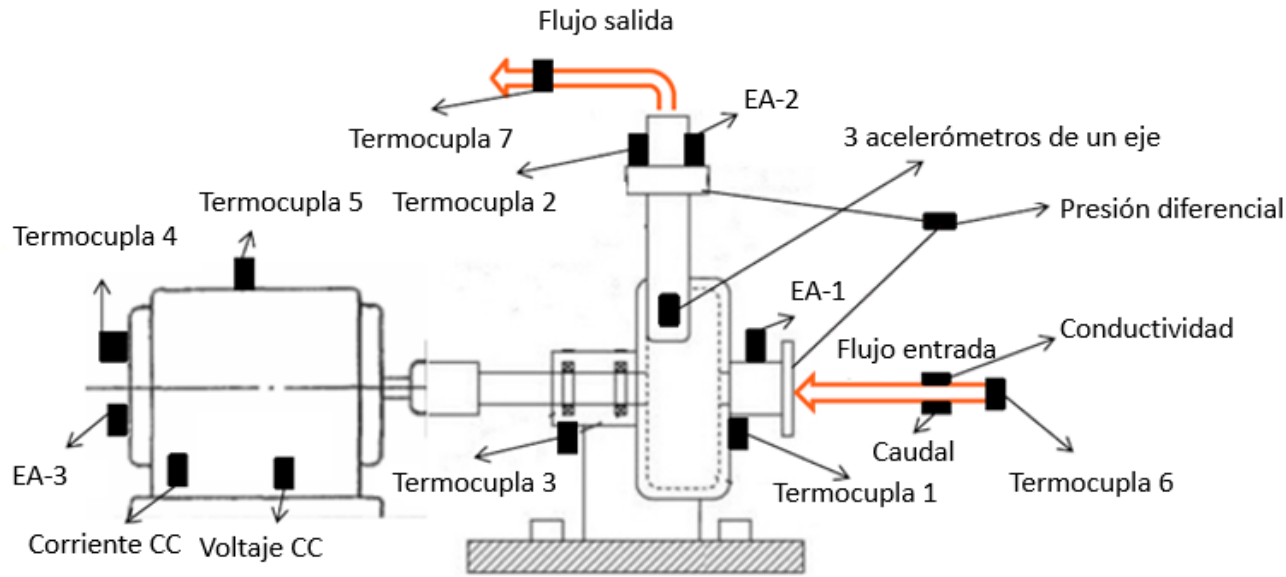


Figura 3.3: Sensores utilizados.

4. Metodología.

La metodología realizada para el desarrollo de este trabajo puede ser vista en la figura 4.1.

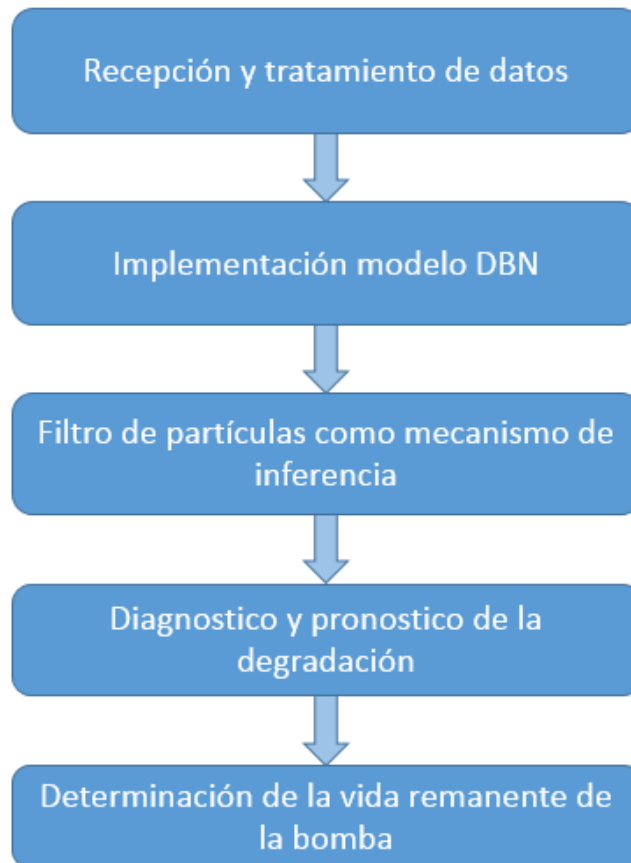


Figura 4.1: Metodología.

Como se indica en la figura anterior, la primera etapa del trabajo consiste en la recepción y el tratamiento de los datos recibidos. Los datos recibidos son datos brutos de los distintos sensores utilizados en el experimento.

Los datos deben ser tratados con el fin de obtener vectores que den cuenta de la degradación que está sufriendo la bomba monitoreada para la implementación del modelo.

Se obtendrán distintos parámetros estadísticos con los datos de vibraciones, presión diferencial entre succión y descarga, caudal, voltaje y corriente para luego correlacionarlas a través de la distancia de Mahalanobis y obtener así una línea de tendencia de la degradación de la bomba.

Una vez realizado el tratamiento de los datos, se procede a la implementación de un modelo en base a redes bayesianas dinámicas, las cuales permiten realizar un monitoreo de carácter dinámico de la bomba estudiada. La figura 4.2 ilustra el modelo a implementar.

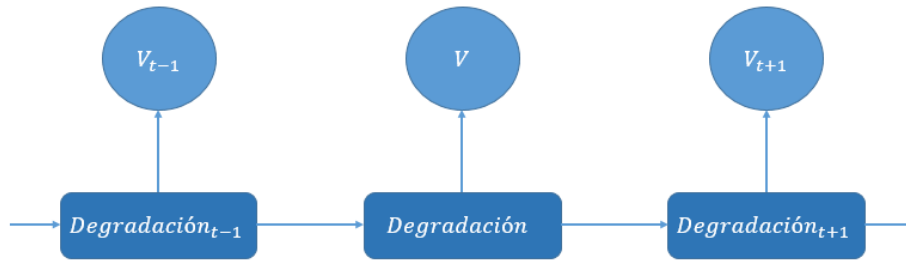


Figura 4.2: Modelo DBN.

La técnica de inferencia de la red, utilizada para diagnosticar y predecir el estado de degradación del sistema en estudio será la técnica de filtro de partículas. Esta técnica consiste en la generación de los distintos escenarios de degradación posibles en cada instante de tiempo según la información disponible, los cuales son ponderados según su probabilidad de ocurrencia para obtener el estado de degradación más probable.

En la implementación del filtro de partículas se utilizan los datos que indican la tendencia de la evolución del estado de degradación de la bomba en el tiempo, para construir los modelos de estado del sistema (system model) y el modelo de medición (measurement model). Para la construcción del modelo de medición se emplea un support vector regression (SVR) para relacionar los datos de vibración con los de eficiencia.

Además de realizar el diagnóstico, es posible realizar el pronóstico de la degradación utilizando la misma técnica, logrando predecir el tiempo de falla de la bomba y el tiempo de vida remanente de la misma. Para obtener dicho tiempo, se realizará la propagación de las últimas partículas generadas, extendiendo y repitiendo el proceso en el tiempo.

La propagación de las partículas permite obtener distintos escenarios de evolución de degradación como se muestra en la figura 4.3:

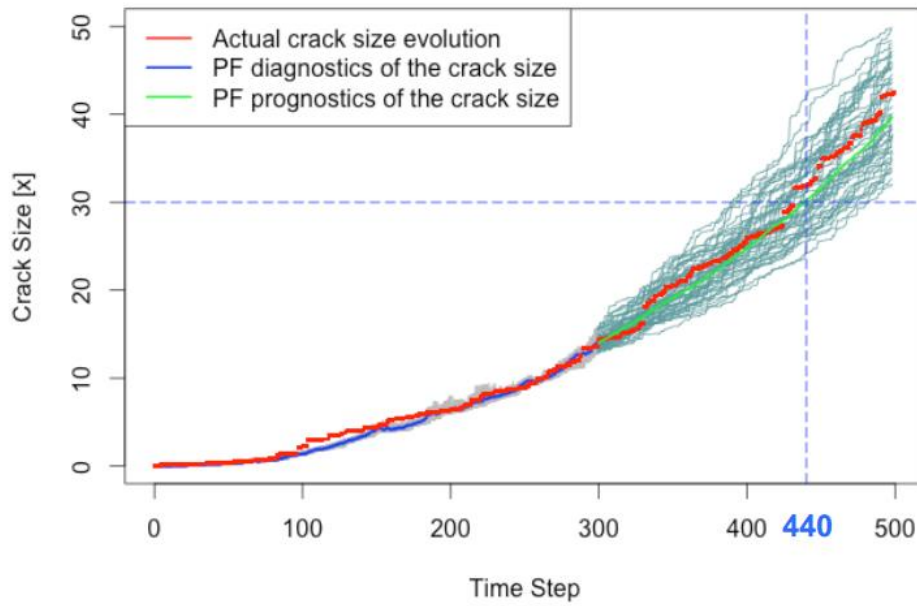


Figura 4.3: Cálculo de tiempo de vida remanente [13].

Luego de la propagación de las partículas en el futuro, se define el criterio de falla. En el caso de la figura, se define como criterio de falla un tamaño de grieta superior a 30. Entonces, para obtener el tiempo de vida de la falla se toma en cuenta la distribución del tiempo en que los distintos escenarios pronosticados cumplen con el criterio y se estima el tiempo de falla según algún criterio, por ejemplo la mediana de los tiempos, la moda, etc.

El tiempo de vida remanente se obtiene entonces como la cantidad de horas entre el instante final de monitoreo y el tiempo de falla estimado.

5. Resultados obtenidos.

Para la implementación del modelo, se realiza primero el tratamiento de datos para la obtención de dos distancias de mahalanobis, una primera construida a partir de los datos de vibraciones y una segunda con los datos de presión diferencial, caudal, corriente y voltaje para observar los cambios de eficiencia de la bomba según se plantea en [14].

5.1. Distancia de mahalanobis en base a vibraciones.

Para la obtención de la primera distancia de mahalanobis, relacionada con los datos de vibraciones, se realiza un análisis espectral utilizando la transformada de Fourier. Los datos de vibraciones contienen información de 3 ejes espaciales (Ejes X, Y, Z). Los datos son separados en intervalos de 1 minuto para realizar el análisis.

Luego de observar diversos espectros obtenidos en distintos intervalos de tiempo para cada eje, se procede a identificar un intervalo de frecuencia de interés y se realiza una división del mismo en 6 bandas de frecuencia, como se propone en [13].

Para la identificación del intervalo de frecuencias se realiza un análisis gráfico del espectro. La figura 5.1 muestra el análisis del espectro para el eje X, el mismo análisis se realiza para los otros dos ejes. Las figuras correspondientes a los ejes Y y Z pueden ser encontradas en el Anexo A.

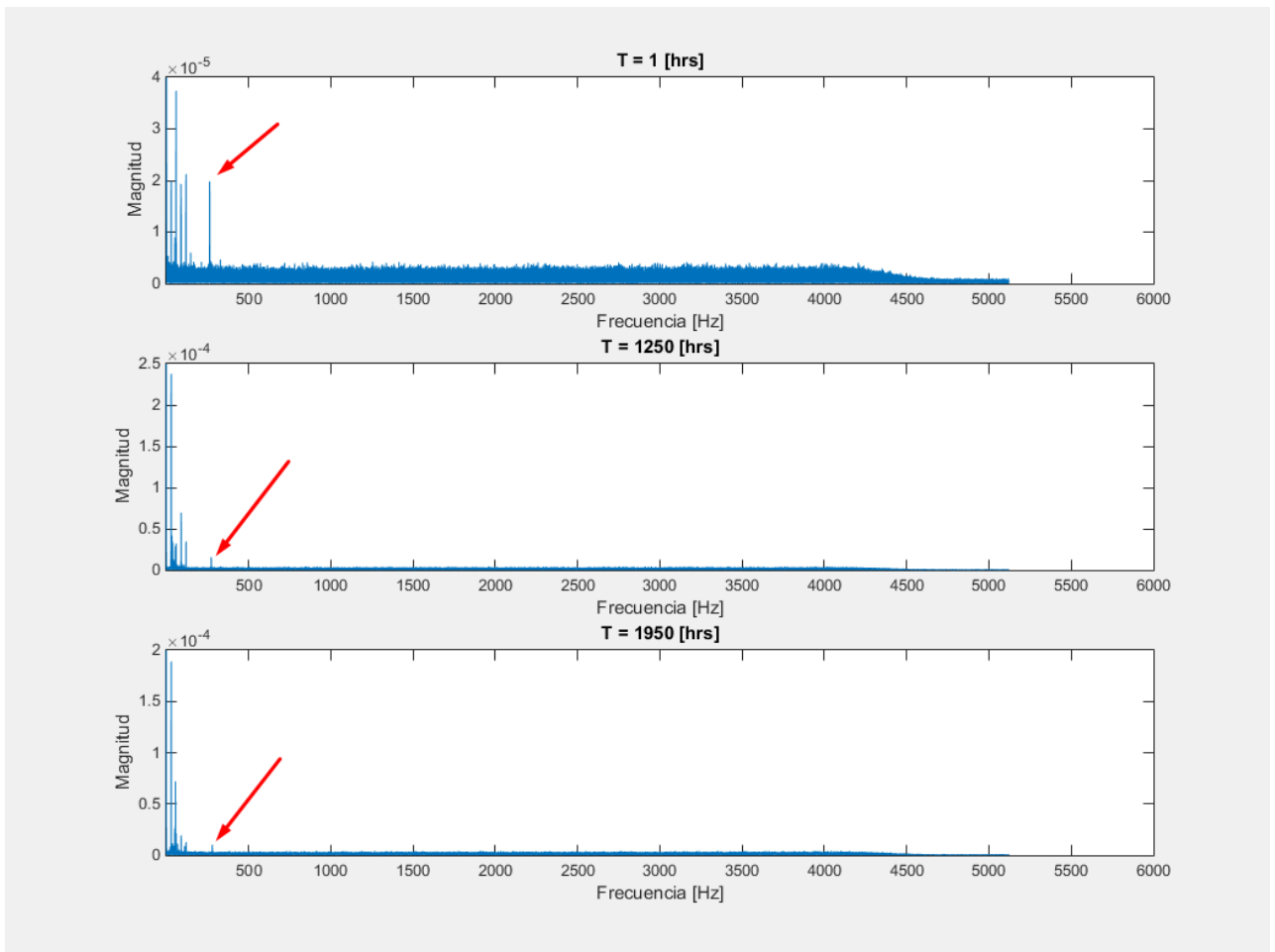


Figura 5.1: Análisis espectral.

De la figura es posible reconocer que para frecuencias mayores a 500 [Hz] no presentan peaks relevantes para el estudio. Al analizar el espectro en el intervalo de 0 – 500 [Hz] se obtienen los siguientes gráficos.

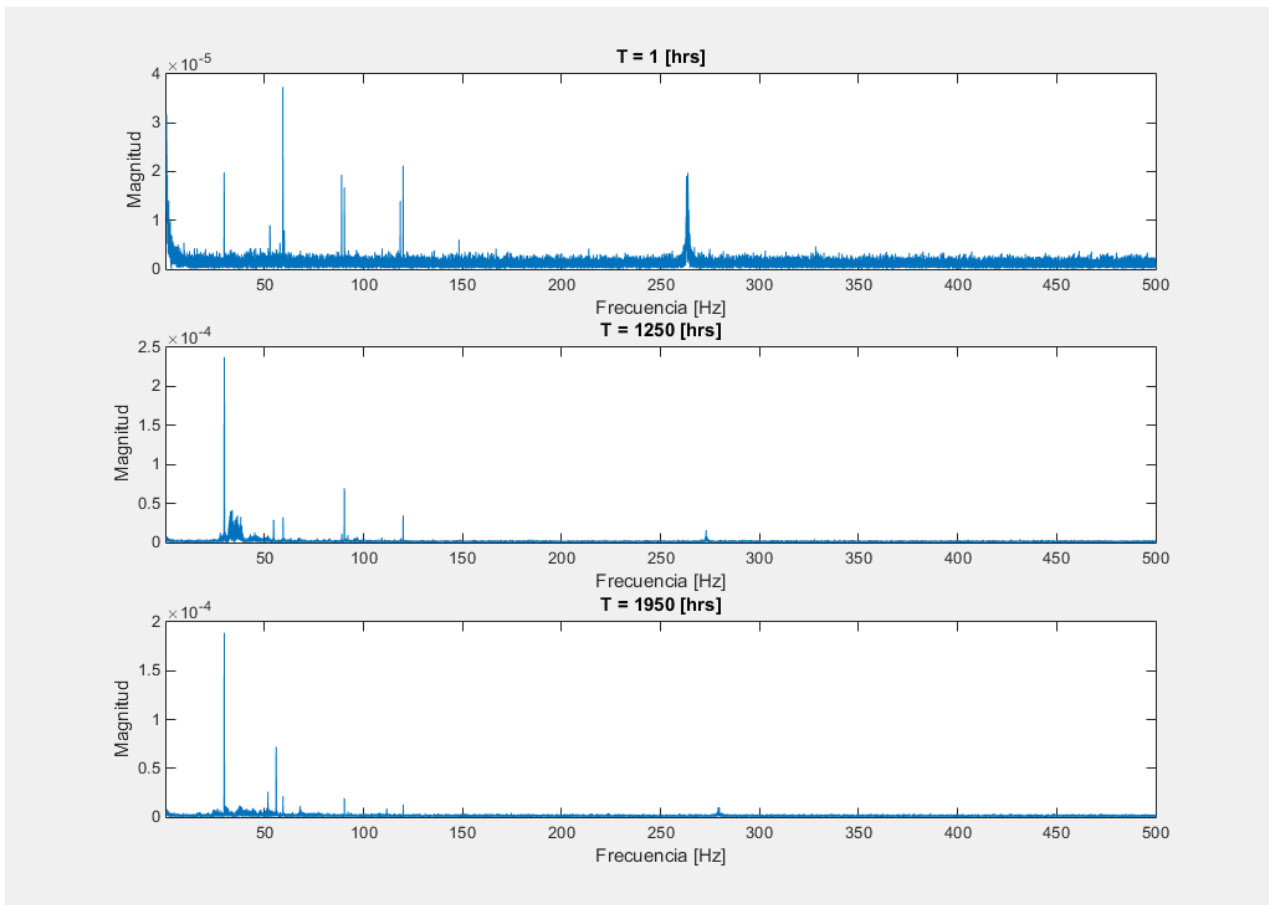


Figura 5.2: Análisis espectral 0 - 500 [Hz]

Luego de este nuevo análisis, se decide acotar el rango de frecuencias de interés al rango de 0 – 300 [Hz] (En realidad, el intervalo es de 0.5 – 300 [Hz] con el fin de evitar que la magnitud del espectro en 0 ensucie los cálculos).

Una vez definido el rango de interés, siguiendo con lo comentado en la sección de metodología, se divide el espectro en 6 bandas de frecuencia. Se establecen seis bandas de frecuencia equiespaciadas, es decir, seis bandas de 50 [Hz] cada una.

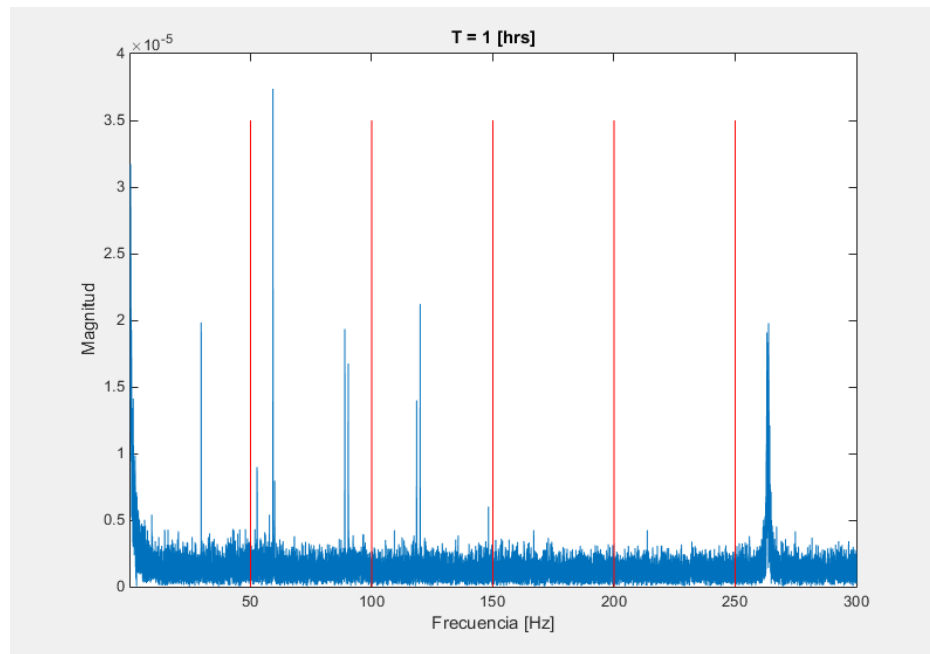


Figura 5.3: Representación bandas.

Una vez realizada la división de bandas, se calcula el valor de la cuadrática media (RMS) de cada una y se construye el vector que agrupe las 18 bandas estudiadas (6 bandas por cada eje) para obtener de esta forma la distancia de mahalanobis a partir de los valores de RMS.

Para calcular la distancia de mahalanobis es necesario identificar una línea base “baseline” con la cual sea posible comparar los datos. Siguiendo los planteamientos descritos en [14], se establece que línea base, que considera la bomba como nueva, los diez primeros días de funcionamiento, es decir, las primeras 240 horas.

Se obtiene entonces la distancia de mahalanobis a partir de los datos de vibraciones. La figura XX muestra la distancia de mahalanobis obtenida.

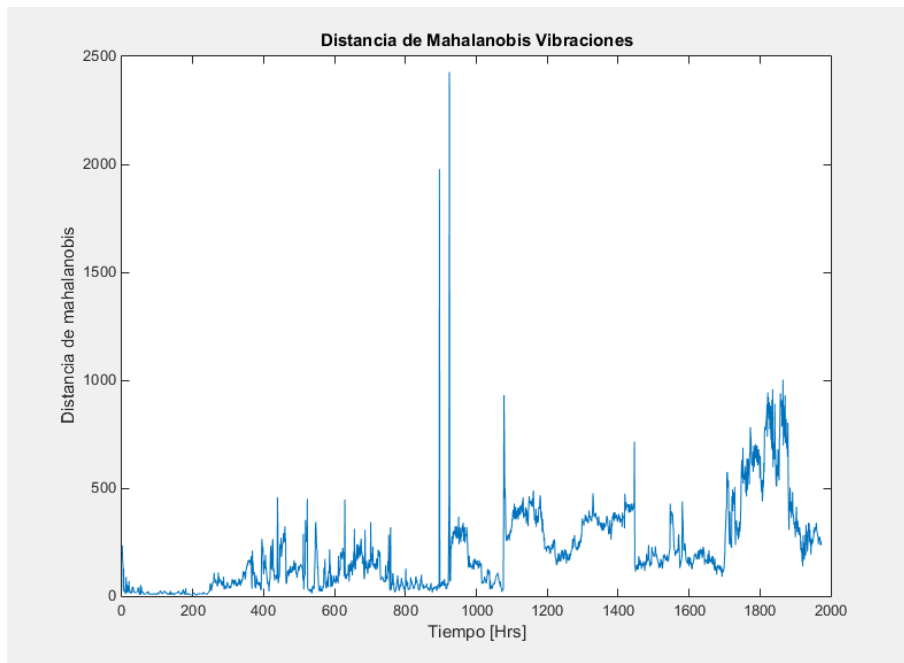


Figura 5.4: Distancia de mahalanobis en base a los datos de vibración.

A simple vista, no es posible identificar de forma fácil una tendencia al resultado, por lo cual es útil observar el resultado de aplicar el logaritmo al resultado anterior.

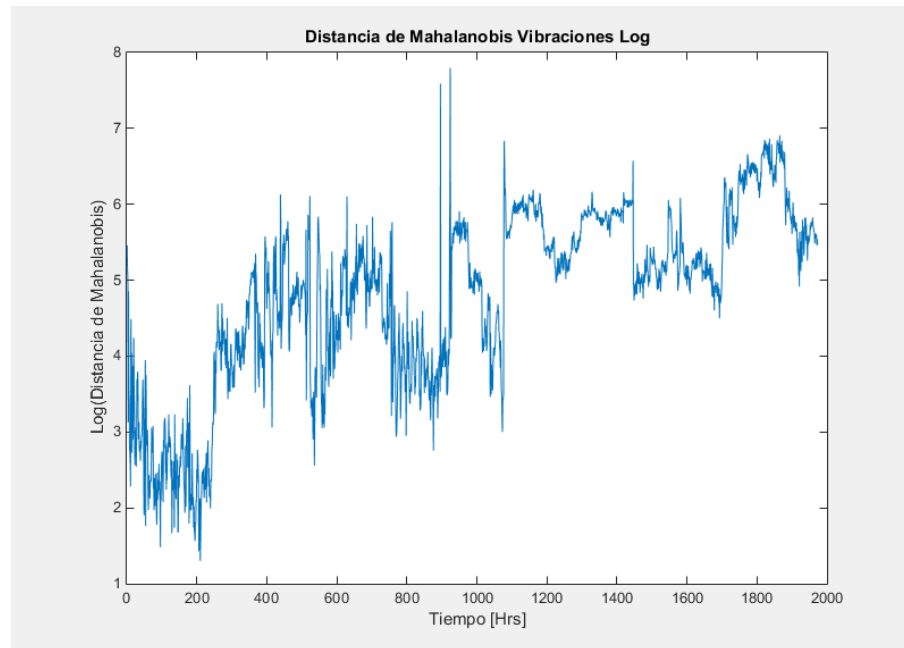


Figura 5.5: Logaritmo de la distancia de mahalanobis.

La figura 5.5 muestra el resultado de aplicar el logaritmo al resultado de la figura anterior (5.4). En esta figura es fácil reconocer una tendencia a un aumento en la distancia de mahalanobis, lo que se traduce en un aumento en las vibraciones de la bomba conforme avanza el tiempo de funcionamiento de esta.

5.2. Distancia de Mahalanobis en base a la eficiencia.

Para la obtención de esta distancia se trabaja con los datos brutos de los sensores que entregan las medidas de presión diferencial entre carga y descarga, caudal, corriente y voltaje. Estas 4 medidas se combinan siguiendo lo planeado por [14] con el fin de obtener una medida de eficiencia de la bomba.

La eficiencia de la bomba se calcula entonces según la ecuación 34.

$$\eta = \frac{PD * C}{V * I} \quad (35)$$

Donde:

- PD = Presión diferencial [psi]
- C = Caudal [lpm]
- V = Voltaje [Vdc]
- I = Corriente [Adc]

Se calculan entonces diferentes parámetros de eficiencia: Media cuadrática (RMS), Valor medio (Mean Value), valor pico a pico (Peak to Peak), desviación estándar (Standard Deviation), factor de cresta (Crest Factor).

Los parámetros obtenidos se muestran en la figura 5.6

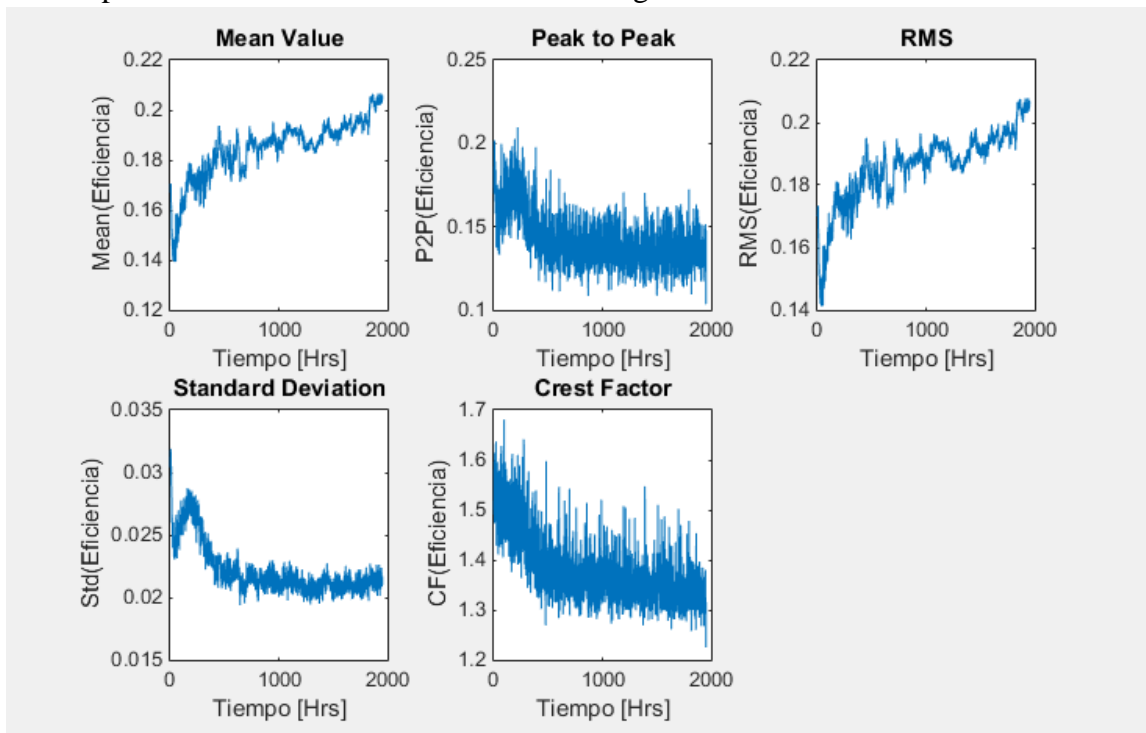


Figura 5.6: Parámetros eficiencia.

Si siguiendo con el criterio de la línea base “baseline” anterior, se consideran los primeros diez días de funcionamiento como la línea base. Con lo anterior es posible obtener la distancia de mahalanobis en base a la eficiencia de la bomba.

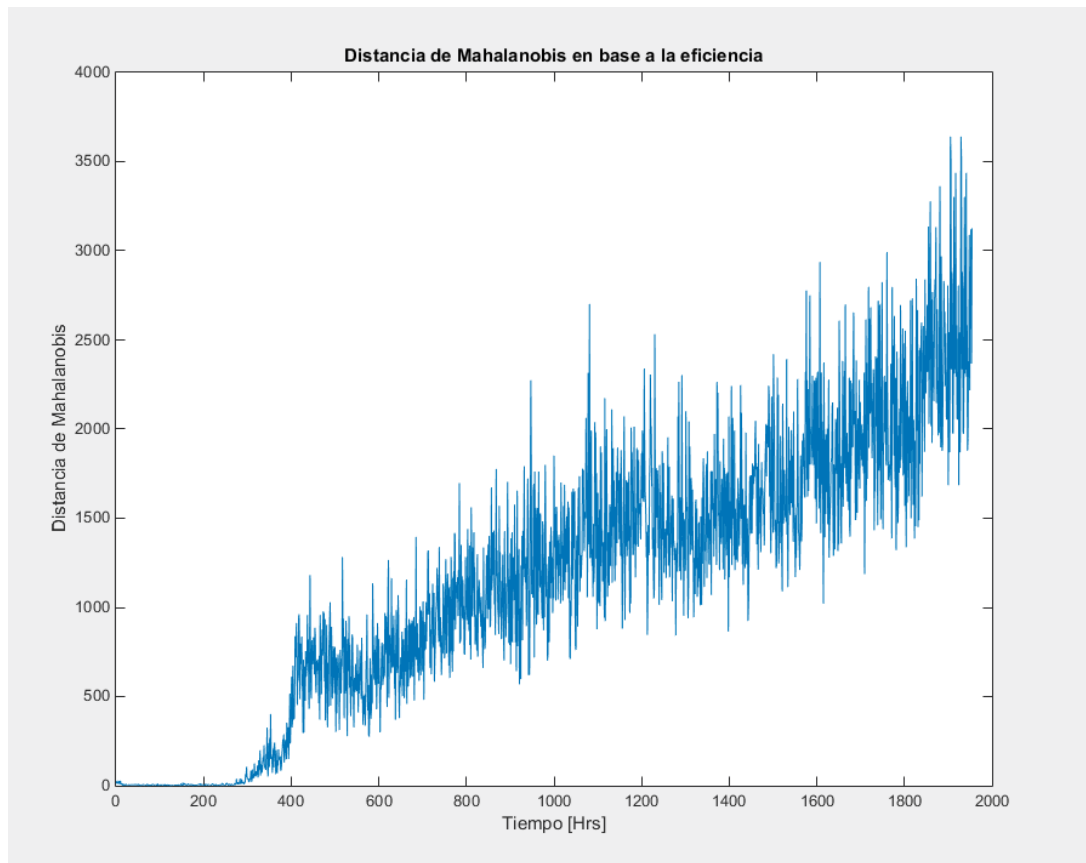


Figura 5.7: Distancia de mahalanobis en base a la eficiencia de la bomba.

En esta oportunidad, es fácil reconocer una tendencia de la distancia. Podemos ver que la distancia en base a la eficiencia va aumentando a medida que aumenta el tiempo de funcionamiento de la bomba, en este caso entonces, la medida de eficiencia debiese ser considerada realmente como una medida de deficiencia de la bomba, término que se utilizará de aquí en adelante en este informe.

5.3. Empirical Mode Decomposition (EMD).

Una vez obtenidas ambas distancias que dan cuenta de los cambios en las vibraciones y la eficiencia de la bomba es necesario buscar alguna forma de sustraer de estas una tendencia más suave para ingresar en el modelo ya que ambas tienen mucho ruido asociado, el cual puede deberse a factores externos o efectos que no nos son de importancia si queremos obtener la tendencia general del comportamiento de la bomba.

La técnica Empirical Mode Decomposition (EMD) permite obtener la tendencia de los datos obtenidos en ambas distancias de mahalanobis. Al realizar la descomposición, el valor del resto de esta indica la tendencia de los datos (ecuación 24).

Las figuras 5.8 y 5.9 muestran las tendencias obtenidas una vez realizada la descomposición de los vectores de deficiencia y vibraciones respectivamente. La función utilizada para la realización del EMD se encuentra en la referencia [16].

En el caso de la tendencia de la deficiencia obtenida, esta fue normalizada para una fácil comprensión, el valor 1 es el valor máximo de deficiencia, en el cual la bomba deja de funcionar.

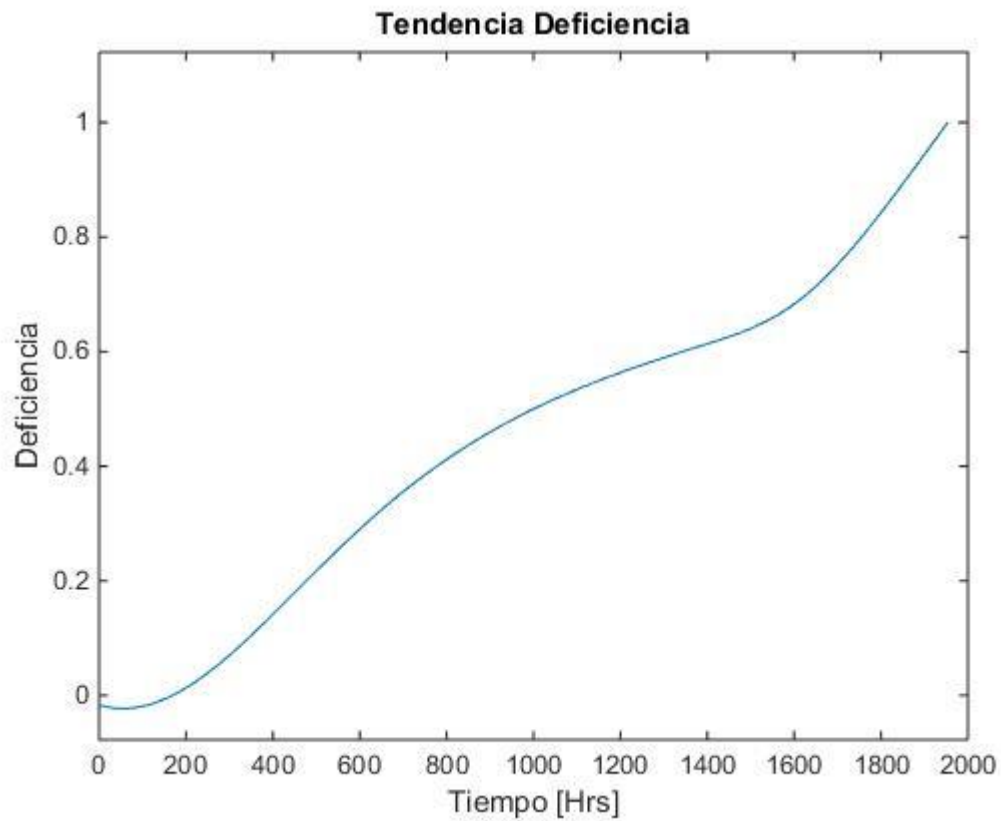


Figura 5.8: Tendencia deficiencia.

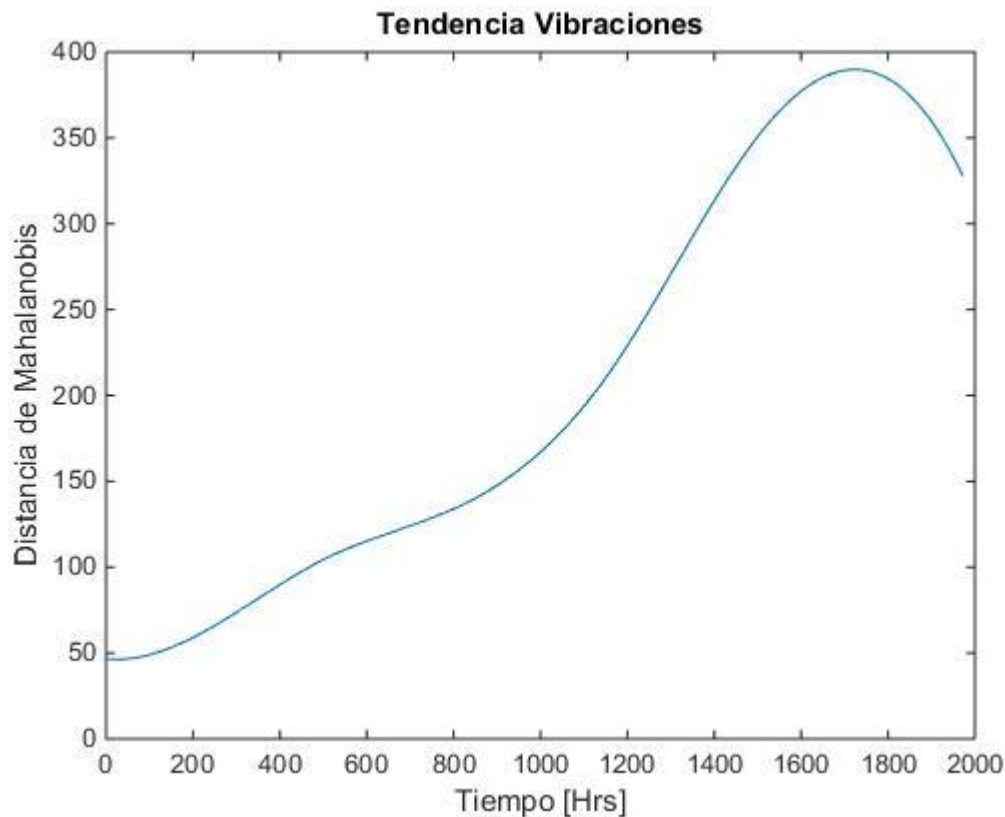


Figura 5.9: Tendencia vibraciones.

Una vez obtenidas ambas tendencias es posible avanzar a la implementación del modelo.

5.4. Implementación del modelo.

Para la implementación del modelo de DBN por medio del filtro de partículas es necesario desarrollar el modelo del sistema, que relaciona el estado actual de la bomba con su estado futuro, y el modelo de medición, que da cuenta de la relación que existe entre el estado interno de la bomba con los datos obtenidos por los acelerómetros. Para la formulación del primero modelo, se realiza una regresión polinomial a la tendencia de eficiencia de la bomba, mientras que en el caso del measurement model, se entrenara un support vector machine (SVM) para encontrar la relación entre las vibraciones y la deficiencia de la bomba.

Debido a que la idea del trabajo es lograr predecir la vida remanente de la bomba, se utilizara un 80% de los datos disponibles, es decir, se dispondrá de los datos hasta el instante $t = 1561$ del total de 1954 horas monitoreadas.

5.4.1. Modelo del sistema (System Model).

Como ya fue explicado, se realiza una regresión polinomial a los datos disponibles. Se escoge una regresión de 4to orden ya que esta se ajusta de forma satisfactoria a la curva. Los resultados de la regresión se observan de forma gráfica en la figura 5.10.

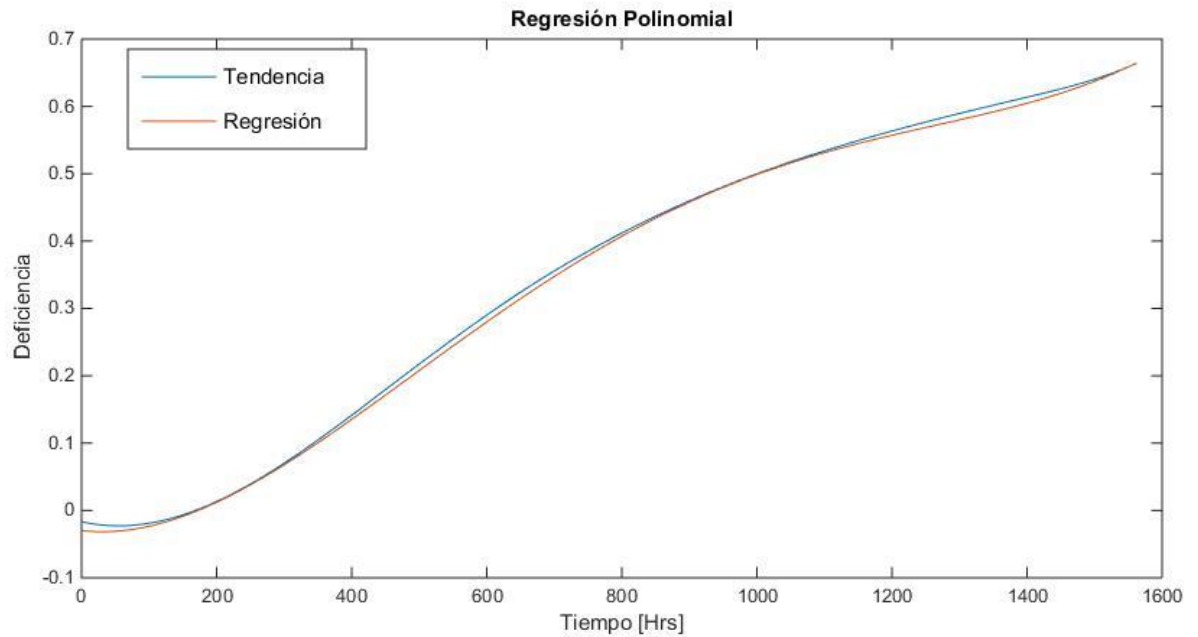


Figura 5.10: Regresión polinomial.

Producto de la regresión se obtiene la siguiente función:

$$z = \frac{t - 781,5}{451,05} \quad (36)$$

$$D = 0.022177 * z^4 - 0.018928 * z^3 - 0.093058 * z^2 + 0.25747 * z + 0.40243 \quad (37)$$

Donde:

$t = \text{tiempo [Hrs]}$

$D = \text{deficiencia}$

Por último, es necesario discretizar la función según [15] con el fin de obtener una relación entre el estado de degradación actual y el estado de degradación siguiente ($D_t \rightarrow D_{t+1}$).

Se obtiene el siguiente system model.

$$D_t = D_{t-1} + (1.9667 * 10^{-4} * z^3 - 1.2589 * 10^{-4} * z^2 - 9.1482 * 10^{-7} * t + 0.0013) * \Delta t * e^{\omega_k} \quad (38)$$

Donde:

$\omega \sim N(0, \sigma_\omega^2)$ Es un ruido blanco gaussiano.

La adición de ruido da cuenta de la naturaleza estocástica intrínseca del modelo.

5.4.2. Modelo de medición (Measurement model).

Para la obtención del modelo de medición, se entrena un support vector regression (SVR) en el software R con el 80% de los datos de vibraciones disponibles. Se entrena en modelo con los datos de la tendencia de la deficiencia como los datos de entrada y los datos de la tendencia de vibraciones como los datos de salida. De esta forma, se espera obtener aproximaciones de las vibraciones de la bomba con los resultados de deficiencia del modelo del sistema (system model). En específico se entrena un SVR con un kernel del tipo RBF. Se obtiene un SVR con 142 vectores de soporte (SV). La figura 5.11 muestra los resultados del SVM.

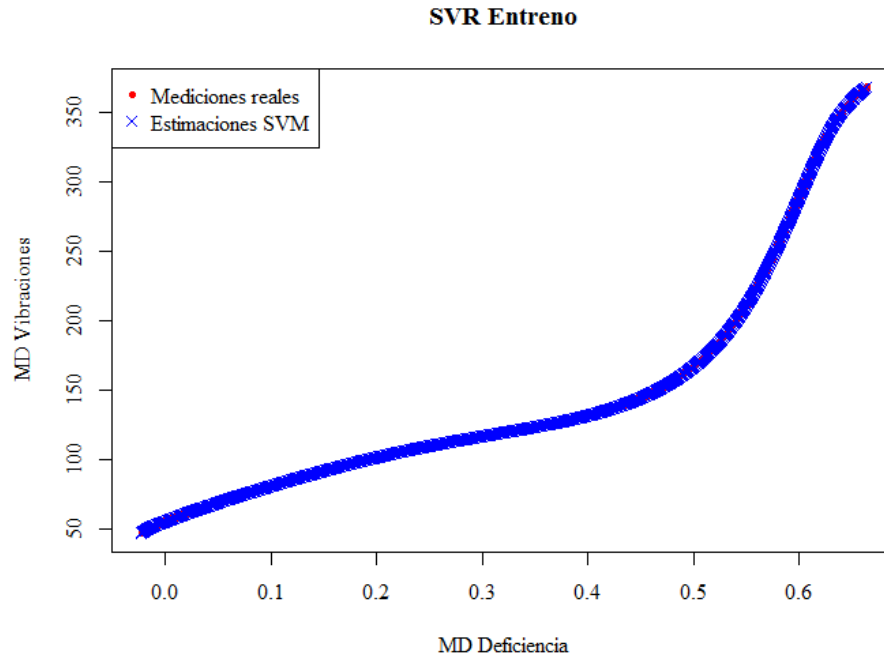


Figura 5.11: Resultados SVM.

5.5. Diagnóstico y pronóstico de la degradación.

Una vez identificado el system model y el measurement model es posible desarrollar el modelo de predicción en base al filtro de partículas, para los cálculos se utiliza una muestra de $N = 50000$ partículas. El resultado del diagnóstico y pronóstico de la degradación se muestra en la figura 5.12.

Predicción Filtro de Partículas

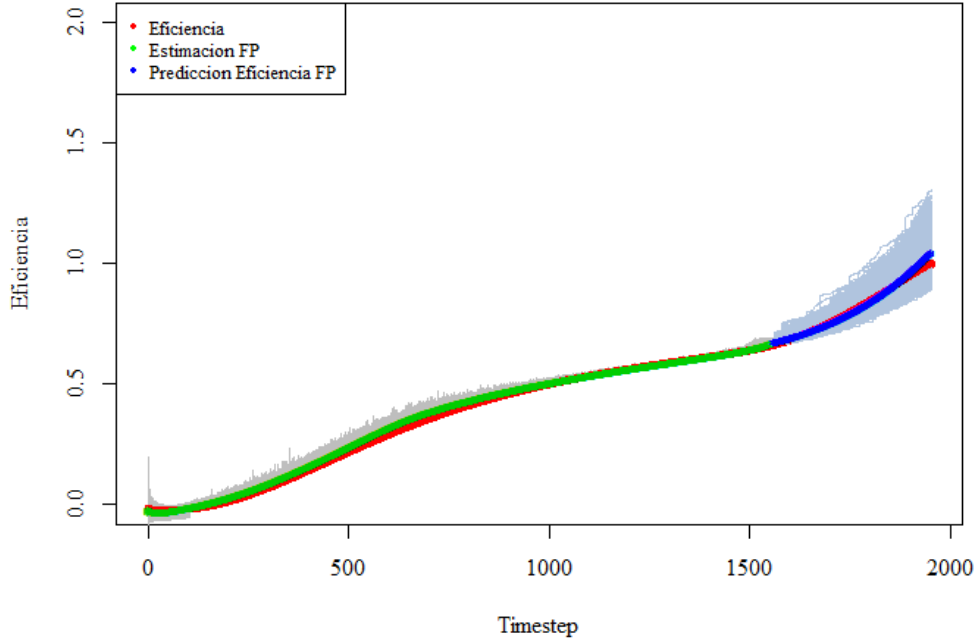


Figura 5.12: Diagnóstico y pronóstico.

5.6. Determinación de la vida remanente de la bomba.

Para la determinación de la vida remanente de la bomba, se define un intervalo de deficiencia cercano entre los valores 1 y 1.01 para observar todas las partículas que se encuentran entre dicho intervalo y luego se realiza el promedio del tiempo de falla (MTTF) de la bomba según la distribución de llegada de las partículas en dicho intervalo a través del tiempo. El resultado se muestra de forma gráfica en la figura 5.13.

Predicción Tiempo De Vida Remanente

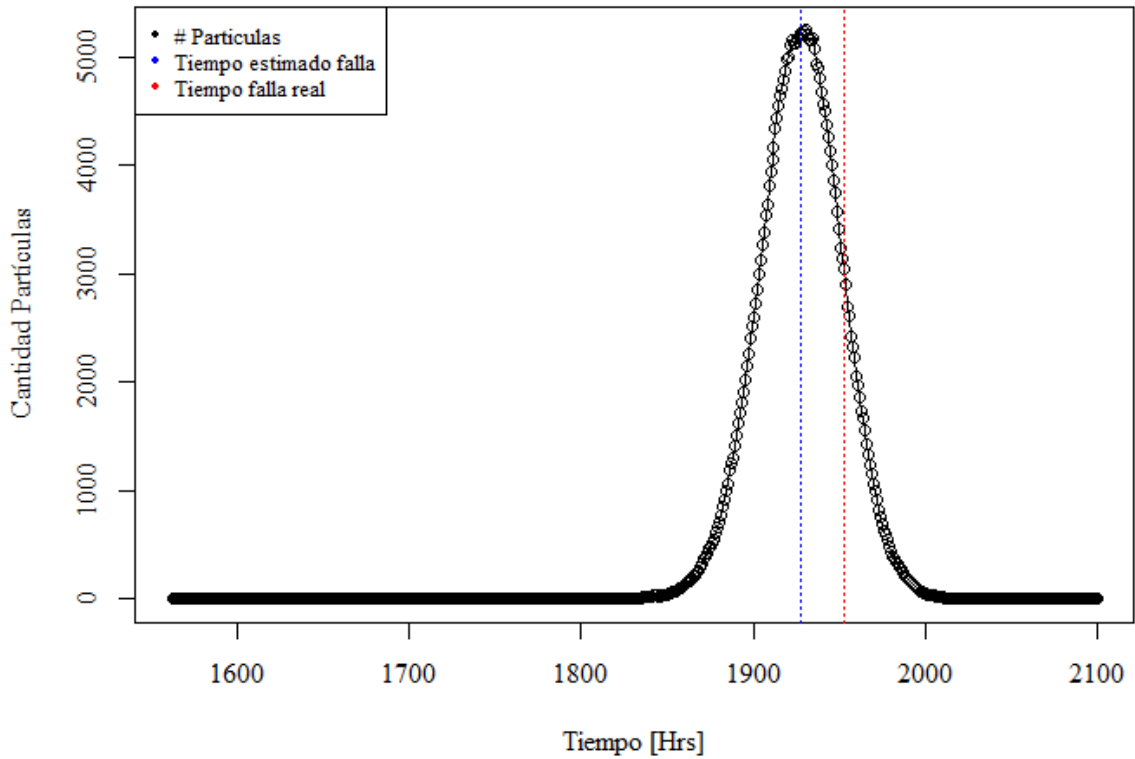


Figura 5.13: Tiempo de vida remanente.

Se obtiene un tiempo de falla de 1927.45 horas, comparado al tiempo de falla real de 1954 horas da una variación de un 1,35% del tiempo real de falla.

El instante en que comienza el pronóstico es en $t = 1561$, con lo que se obtiene un tiempo de vida remanente de 366.45 horas o alrededor de 15 días.

El detalle de los códigos desarrollados puede ser consultado en la sección B de los anexos.

6. Discusión y conclusiones.

Parte importante del desarrollo de este trabajo dice relación con el tratamiento de los datos brutos, siendo una etapa importante y determinante para lograr buenos resultados. La obtención de parámetros que puedan dar cuenta de la evolución del estado de bomba es indispensable para la predicción del tiempo de vida remanente y en caso de no identificar ninguna feature adecuada no es posible realizar la predicción.

El modelo del sistema (system model) fue desarrollado utilizando información de la deficiencia de la bomba ya funcionando, a partir de los datos obtenidos, lo cual restringe la exactitud de la predicción. Debido a la naturaleza de este modelo, la cantidad de datos utilizados para su desarrollo está directamente relacionado a la calidad de la predicción. En caso de utilizar una menor cantidad de datos, por ejemplo 50% de estos, la predicción no habría sido correcta. Para una mayor libertad en este sentido sería correcto profundizar en el mecanismo de obtención del system model, como por ejemplo en [1], donde se utiliza un modelo empírico y la técnica de “*joint particle filtering*”[1], sin embargo, el caso de estudio de este informe es un sistema completo (bomba) de la cual no existe información de modelos empíricos, por lo cual el enfoque utilizado en este estudio es satisfactorio para el estudio específico de la bomba monitoreada, pero puede no ser utilizado para el estudio de otras, debido que se desarrolló para ajustarse a los datos obtenidos de la bomba estudiada.

Los resultados obtenidos son satisfactorios: Se cumplen los objetivos y el alcance del trabajo de título realizado, obteniendo una predicción satisfactoria y un tiempo de vida remanente de 366.45 horas o 15 días con una diferencia mínima de un 1,35% entre el tiempo de falla real y el tiempo de falla estimado por el alumno.

7. Bibliografía.

- [1] Rabiei, E.; Droguett, E L; Modarres, M. A, “Prognostics Approach Based on the Evolution of Damage Precursors using Dynamic Bayesian Networks”, *Advances in Mechanical Engineering*, v. 8(9), p. 1 – 19, 2016.
- [2] Kevin Patrick Murphy, “Dynamic Bayesian Networks: Representation, Inference and Learning”, University of Berkeley, 2002.
- [3] M.E. Orchard, G.J. Vachtsevanos, “A Particle-Filtering Approach for On-Line Fault Diagnosis and Failure Prognosis”, *Transactions of the Institute of Measurement and Control*, v.31, 2009.
- [4] K. B. Korb, A. E. Nicholson, “Bayesian Artificial Intelligence”, Second Edition, CRC Press, 2011.
- [5] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking”, *IEEE Transactions on Signal Processing*, v.50, n.2, 2002
- [6] Zhu J, He D, Bechhoefer E. Survey of lubrication oil condition monitoring, diagnostics, and prognostics techniques and systems. In *Technical Program for MFPT 2012, The Prognostics and Health Management Solutions Conference - PHM: Driving Efficient Operations and Maintenance*. 2012.
- [7] MathWorks - Makers of MATLAB and Simulink, <https://www.mathworks.com>
- [8] De Maesschalck, Roy; Jouan-Rimbaud, Delphine; and Massart, Désiré L. (2000); The Mahalanobis distance, *Chemometrics and Intelligent Laboratory Systems* 50:1–18
- [9] Lambert, Max; Engroff, Andrew; Dyer, Matt; Byer, Ben. "Empirical Mode Decomposition".
- [10] C. B. S. Maior, M. C. Moura, I. D. Lins, E. López Droguett and H. H. Diniz , Remaining Useful Life Estimation by Empirical Mode Decomposition and Support Vector Machine, *IEEE LATIN AMERICA TRANSACTIONS*; 14, 11; 4603-4610.
- [11] Huang, N. E.; Shen, Z.; Long, S. R.; Wu, M. C.; Shih, H. H.; Zheng, Q.; Yen, N. C.; Tung, C. C.; Liu, H. H. (1998). "The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Nonstationary Time Series Analysis" (pdf). *Proceedings of the Royal Society of London A*. 454 (1971): 903–995. doi:10.1098/rspa.1998.0193.
- [12] M.D. Moura, E. Zio, I. D. Lins, E. Droguett, Failure and reliability prediction by support vector machines regression of time series data, *Reliability Engineering System Safety* 99 (11)(2011)1527–1534.
- [13] Droguett, E L, *Apuntes ME709-1 Gestión de Activos Físicos Aplicada*, 2016.

[14] T. Zhou, E. Droguett, M. Modarres , A Hybrid Probabilistic Physics of Failure Pattern Recognition based Approach for Assessment of Multi-Unit Causal Dependencies, ICONE24, June 26-30, 2016, Charlotte, North Carolina.

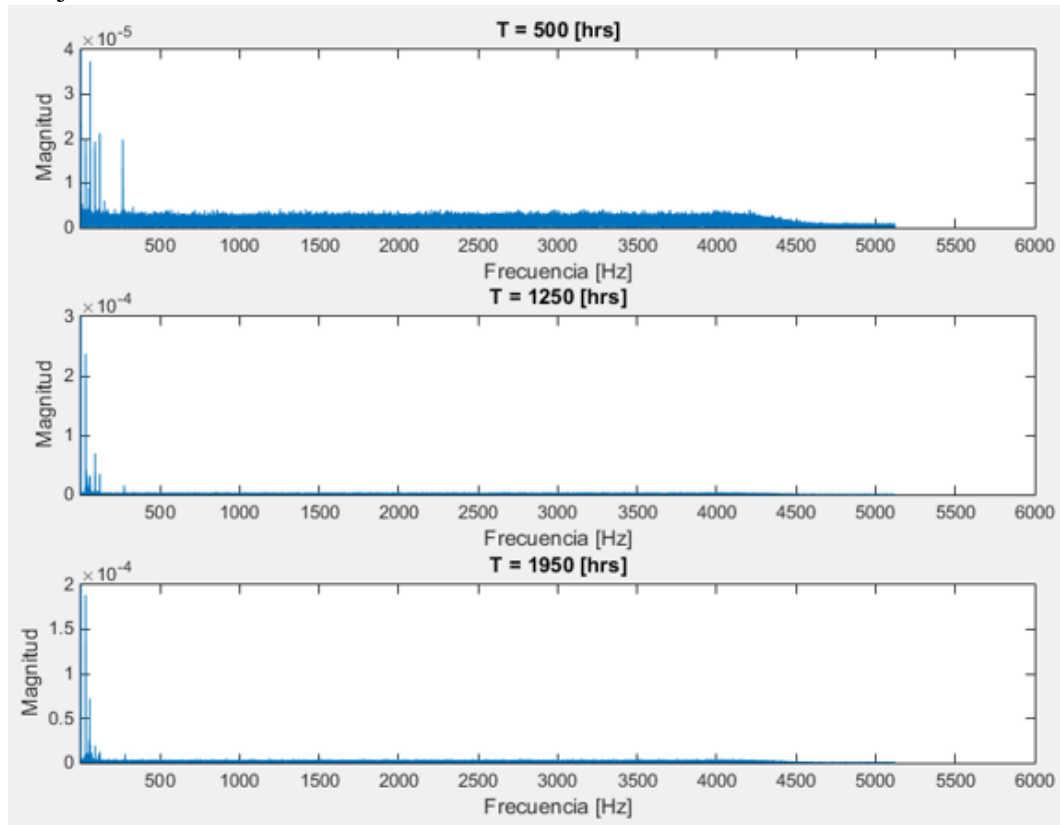
[15] Zio E and Pelsoni G. Particle filtering prognostic estimation of the remaining useful life of nonlinear components. Reliab Eng Syst Safe 2011; 96: 403-409.

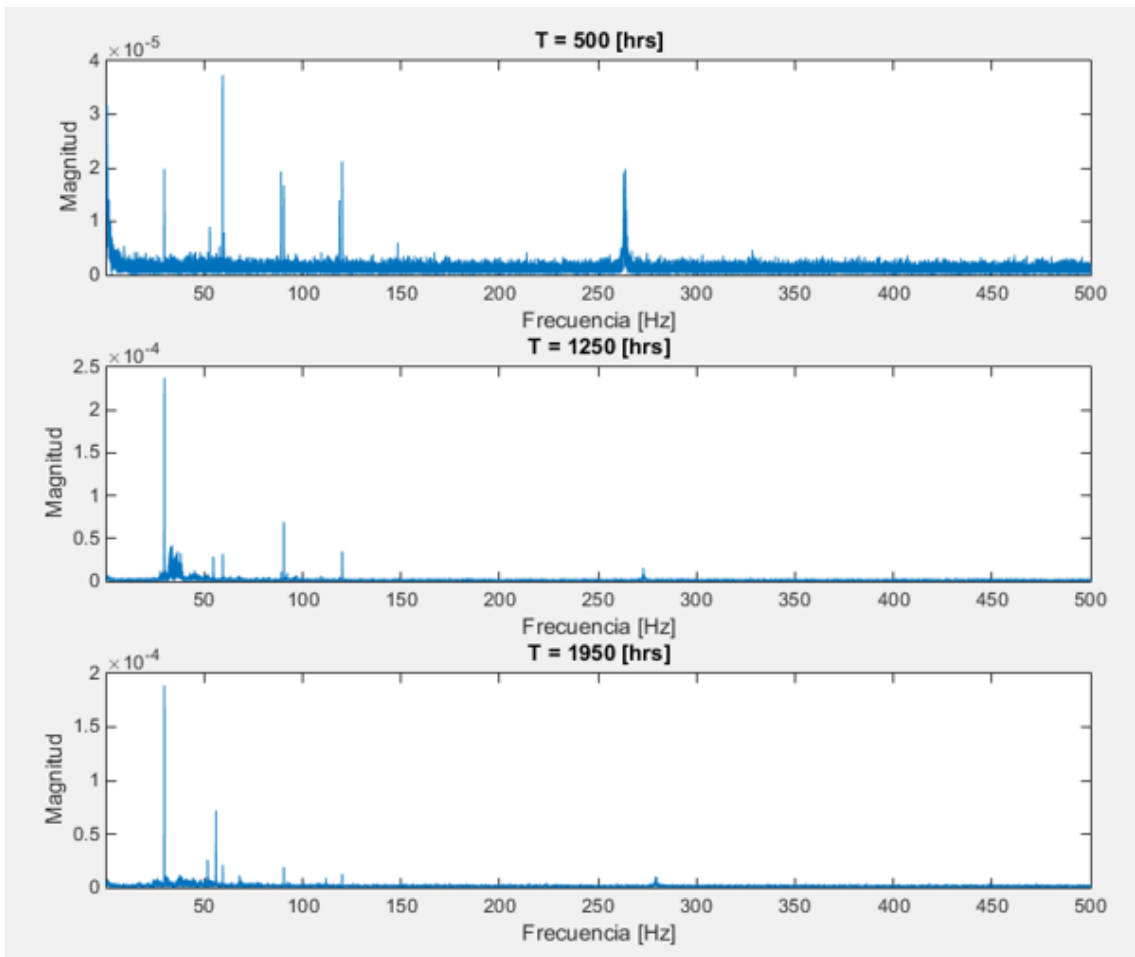
[16] Manuel Ortigueira, <https://www.mathworks.com/matlabcentral/fileexchange/21409-empirical-mode-decomposition>

Anexos.

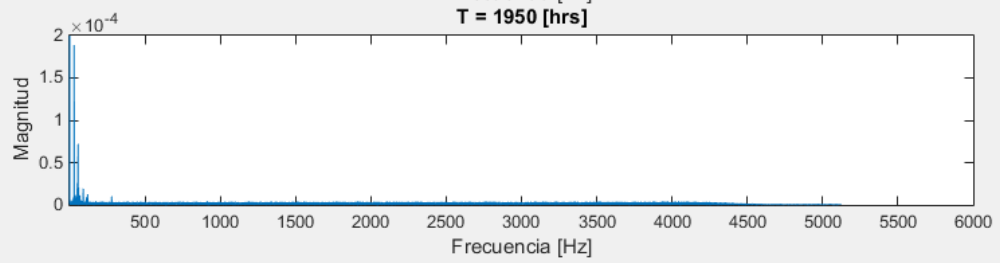
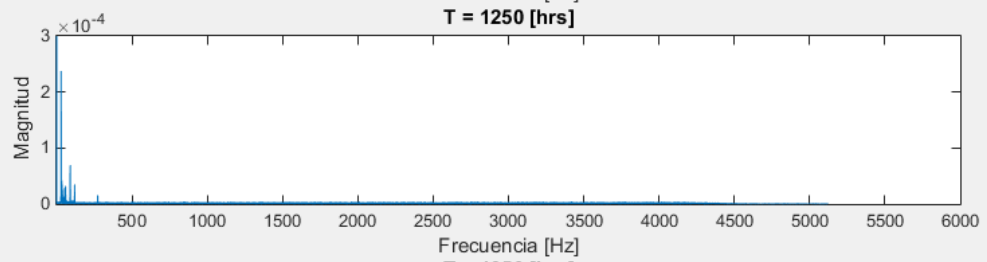
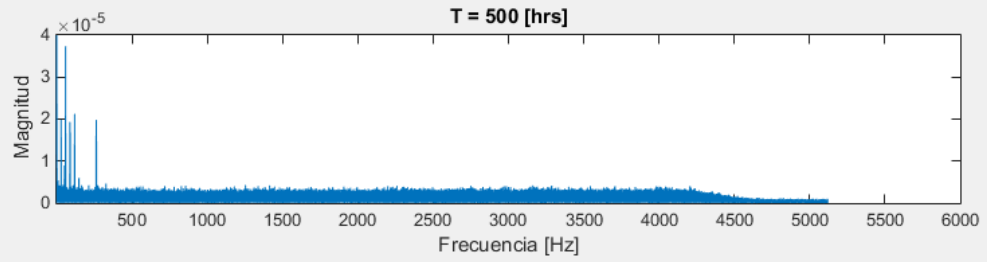
Anexo A

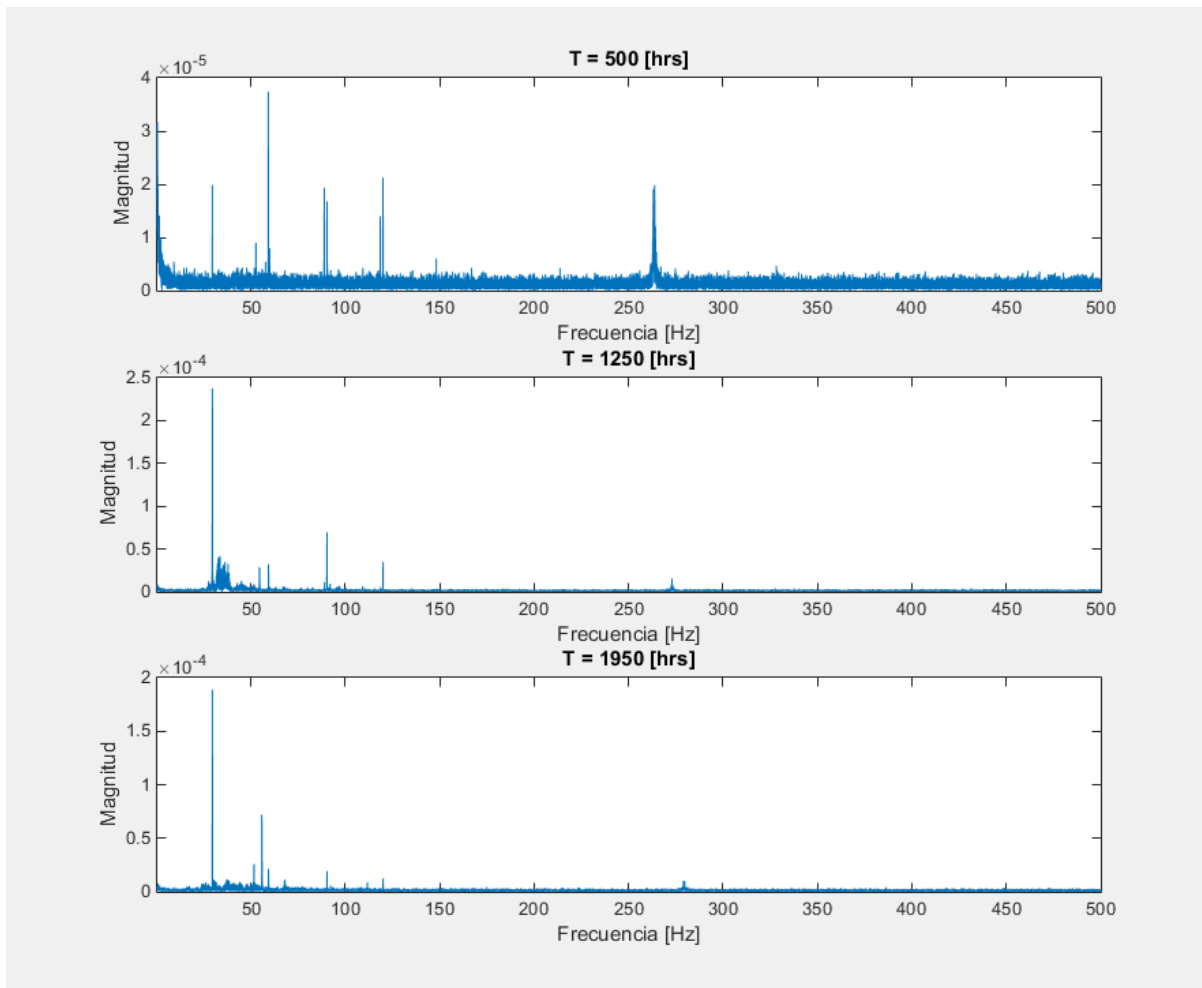
Espectros Eje Y.





Espectros Eje Z.





Anexo B. Códigos.

MATLAB: Obtencion de RMS bandas.

```

%directory_name = uigetdir;
files = dir('D:\Documents\Universidad\2017-01\Data F\UMD Pump 1 Vib\data');
fileIndex = find(~[files.isdir]);
%Características
Fs = 10240;          % Sampling frequency
T = 1/Fs;           % Sampling period
L = Fs*60;          % Length of signal
t = (0:L-1)*T;      % Time vector

freq_min = 1;
freq_max = 300; %Frecuencia maxima de importancia a estudiar.
n = 6; %Número de bandas.

%Features; RMS
RMSX = zeros(1,n) ;RMSY = zeros(1,n);RMSZ = zeros(1,n);
%RMSX_f = zeros(1,n);RMSY_f = zeros(1,n);RMSZ_f = zeros(1,n);
%CrestF_X = zeros(1,n);CrestF_Y = zeros(1,n);CrestF_Z = zeros(1,n);
FFTX = [];FFTY = [];FFTZ = [];
for i = 1:50:length(fileIndex)

```

```

%for i = 1:1
fileName = files(fileIndex(i)).name;
directory = strcat('D:\Documents\Universidad\2017-01\Data F\UMD Pump 1
Vib\Data', '\', files(fileIndex(i)).name);
DataI = csvread(directory);
%Aplicar Fourier a la data.
for j = 1:3
    Fft = fft(DataI(:,j));
    P2= abs(Fft/L);
    P1 = P2(1:L/2+1);
    %P1(2:end-1) = 2*P1(2:end-1);
    f = Fs*(0:(L/2))/L;
    %Dividir en bandas.
    index1 = find(f==freq_min,1);
    index2 = find(f==freq_max,1);
    ff = f(index1:index2); %Cortamos segun freq min y max
    nn = n-mod(numel(ff),n);
    f2 = [ff, nan(1,nn)]; %agregar NaN para hacer el vector divisible por n.
    fdivided = reshape(f2, [],n); %dividir el vector en n columnas.
    P11 = P1(index1:index2); %Cortamos segun freq min y max
    P12 = ([transpose(P11), nan(1,nn)]);
    Pdivided = reshape(P12, [],n);
    %sacar ultima columna
    x = Pdivided(:,n);
    xx = x(find(isfinite(x) & x ~= 0));
    Pdivided = Pdivided(:,1:n-1);
    Rms = [rms(Pdivided), rms(xx)];
    CrestF = [peak2rms(Pdivided), peak2rms(xx)];
    if j == 1
        FFTX = [FFTX, P1];
        RMSX = [RMSX; Rms];
    elseif j == 2
        FFTY = [FFTX, P1];
        RMSY = [RMSY; Rms];
    elseif j == 3
        FFTZ = [FFTX, P1];
        RMSZ = [RMSZ; Rms];
    end
end
end
end

```

MATLAB: MD bandas RMS.

```

files = dir('D:\Documents\Universidad\2017-01\Data F\UMD Pump 1 Vib');
dirX = strcat('D:\Documents\Universidad\2017-01\Data F\Codigos\CSV
Pump1\Pump1_RMSX_6band_1_300.csv');
dirY = strcat('D:\Documents\Universidad\2017-01\Data F\Codigos\CSV
Pump1\Pump1_RMSY_6band_1_300.csv');
dirZ = strcat('D:\Documents\Universidad\2017-01\Data F\Codigos\CSV
Pump1\Pump1_RMSZ_6band_1_300.csv');

RMSX = csvread(char(cellstr(dirX)));
RMSY = csvread(char(cellstr(dirY)));
RMSZ = csvread(char(cellstr(dirZ)));

VectorXYZ = [];
Baseline = [];

```

```

%10 dias como baseline. 24*10
for i = 1:24*10
RMS_i = [RMSX(i,:),RMSY(i,:),RMSZ(i,:)];
Baseline = [Baseline;RMS_i];
end

for i = 1:length(RMSX(:,1))
Vectori = [RMSX(i,:),RMSY(i,:),RMSZ(i,:)];
VectorXYZ = [VectorXYZ;Vectori];
end
MD_VD_Pump1 = mahal(VectorXYZ,Baseline);
[~,filas] = size(VectorXYZ);
MD_VD_Pump1_div = MD_VD_Pump1./filas;

clear dirX dirY dirZ RMS_i RMSX RMSY RMSZ Vectori i ii files
clear filas VectorXYZ

```

MATLAB: Parámetros eficiencia.

```

files = dir('D:\Documents\Universidad\2017-01\Data F\Eficiencia\Pump 1_Raw
Efficiency Data\data ordenada');
fileIndex = find(~[files.isdir]);

% FLOW RATE [LPM] - DC VOLTAGE [Vdc] - DC CURRENT [Adc] - DIFFERENTIAL
% PRESSURE [PSI]

%DIA;TIEMPO;FR;DCV;DCC;DP
Features = [];
for index = 2:length(fileIndex)
%for index = 3:3
%for index = 7
    fileName = files(fileIndex(index)).name;
    directory = strcat('D:\Documents\Universidad\2017-01\Data
F\Eficiencia\Pump 1_Raw Efficiency Data\data
ordenada','\',files(fileIndex(index)).name);
    excel = xlsread(directory);
    tiempo_inicio = datestr(excel(1,1) + 693960, 13); %tiempo inicio excel.
    hora_inicio = str2num(tiempo_inicio(1:2));
    indice_inicio = 1;
    horas = 0;
    indices_horas = [];
    for i = 1:length(excel) %Buscar los indices de cambios de hora.
        time = datestr(excel(i,1) + 693960, 13); %conversion de tiempo actual
exponencial a HH:MM:SS.
        hora_actual = str2num(time(1:2)); %La hora de los datos actuales.
        if str2num(time(1:2)) ~= hora_inicio %distinto a.
            horas = horas + 1;
            hora_inicio = str2num(time(1:2));
            indices_horas = [indices_horas,i];
        end
    end
    indices_horas = [1,indices_horas,i];
    %Construcción de features.
    % Mean Value; Peak to Peak; RMS; Standard Deviation; Crest Factor;
    Features_dia = [];
    for i = 2:length(indices_horas)
        Mean_V = mean(excel(indices_horas(i-1):indices_horas(i),6));
    end

```

```

        P2P = peak2peak((excel(indices_horas(i-1):indices_horas(i),6)));
        RMS = rms((excel(indices_horas(i-1):indices_horas(i),6)));
        Std = std(excel(indices_horas(i-1):indices_horas(i),6));
        Crest_F = peak2rms(excel(indices_horas(i-1):indices_horas(i),6));
        Features_i = [Mean_V,P2P,RMS,Std,Crest_F];
        Features_dia = [Features_dia;Features_i];
    end
    indices_horas(i);
    Features = [Features;Features_dia];
end

```

MATLAB: MD Eficiencia y EMD.

```

files = dir('D:\Documents\Universidad\2017-01\Data F\Eficiencia\Pump 1_Raw
Efficiency Data');
Feat = strcat('D:\Documents\Universidad\2017-01\Data F\Eficiencia\Pump 1_Raw
Efficiency Data\Features_Pump_1.csv');
mahalis = [];

Features = csvread(Feat);
%Mean_V = Features(:,1);
%P2P = Features(:,2);
%RMS = Features(:,3);
%Std = Features(:,4);
%Crest_F = Features(:,5);
%10 días como baseline. 24*10
Baseline = Features(1:24*10,:);
MDP1 = mahal(Features,Baseline);
MDP1_div = MDP1./5;

Res1 = 50;
Res2 = 50;
EMD_P1 = rParabEmd__L(MDP1,Res1,Res2,1);
EMD_P1_div = rParabEmd__L(MDP1_div,Res1,Res2,1);

```

R: Filtro Partículas.

```
rm(list=ls())
library(kernlab)
#C:/Users/Sebastian/Desktop/Filtro R
eff_trend <- read.csv(file="C:/Users/Sebastian/Desktop/Filtro R/Pump1/Eficiencia/Trend80Eff_N.csv",
header=TRUE,
sep=",")
vib_trend <- read.csv(file="C:/Users/Sebastian/Desktop/Filtro R/Pump1/Vibraciones/Trend80Vibra.csv",
header=TRUE,
sep=",")
#Entrenar SVM con un 80% de los datos.
#sv = ksvm(training data eficiencia", vector de respuesta sensores "Vibra")
sv = ksvm(as.matrix(eff_trend),as.matrix(vib_trend),kernel='rbfdot', C = 10000, scaled=c())
predictedY <- predict(sv, eff_trend)
eff_trend_full <- read.csv(file="C:/Users/Sebastian/Desktop/Filtro
R/Pump1/Eficiencia/Trend_MD_P1_Eficiencia_N.csv",
header=TRUE, sep=",")
Eff = as.vector(t(eff_trend))
vib = as.vector(t(vib_trend))
op <- par(family = "serif")
legend("topleft", legend= c("Mediciones reales", "Estimaciones SVM"), pch= c(20, 4), col=
c("red", "blue"),cex=0.5)
plot(Eff,vib,xlab= "MD Deficiencia", ylab = "MD Vibraciones",col="red",pch=20)
points(Eff,predictedY, col = "blue", pch=4)
op <- par(family = "serif")
legend("topleft", legend= c("Mediciones reales", "Estimaciones SVM"), pch= c(20, 4), col=
c("red", "blue"),cex=0.5)
title("SVR Entreno")
#Deficiencia se asume como el "real".
#Deficiencia es process model
#SVR es measurement model (MD vibraciones)
#Las observaciones son los datos de MD vibration.
##----- Generate Observation data-----
Effull = as.vector(t(eff_trend_full))
x.true = Effull #TENEMOS DATA DEFICIENCIA COMO DAÑO REAL?
z.obs <- vib #Datos vibraciones real
##-----
---
##-----Starting particle filter-----
####
N = 10000 # Numero particulas.
delta.t <- 1 # Delta tiempo 1 hora
x.0 <- -0.0255 # eficiencia inicial
#Lfinal <- 1953 # Total number of timesteps CANTIDAD DE HORAS
Lfinal <- 1953
L <- 1561 # number of timesteps that monitoring (diagnosis) is performed (with measurement)
timestep <- seq(0, by= delta.t, length.out= Lfinal)
#x.true = numeric() # numeric vector for generating actual eficiencia
#x.true[1] <- x.0
set.seed(132)
x <- runif(N, min = -0.2, max = 0.2) # Generating N particles uniformly distributed around the
initial deficiency
state.mat <- x # Save the particles' states in a matrix for later plotting
p.weight <- rep(1/N, times = N) # Assign equal weights to all the particles
w.mat <- p.weight # save particle weights in a matrix for later use
xest.out <- x.0 # xest.out is the vector consist of final PF estimation of
the deficiency at each timestep
```

```

tau <- 7.4 #Measurement Noise 2% del max de vib
#sigma <- 774.1764
sigma <- 1 #Process noise (System Model)
for (i in 2:L){
print(i)
#DX Normalizado
dx <- (31960425259582555*((20*i)/9021 - 5210/3007)^3)/162507888954036977664 - (1183*((20*i)/9021 -
5210/3007)^2)/9396875 - (83819194824768725*i)/91623979140897973469184 +
49085763944500680994349/38176657975374155612160000
#print(dx)
ruido <- exp(rnorm(N, mean = 0, sd = sigma))
x.update <- x + dx*delta.t*ruido
#Measurement Model(deficiencia) -> vibracion
z <- predict(sv,x.update) + rnorm(N, mean = 0, sd = tau)
z <- z[1:N]
#----- particle weights-----#
p.weight <- p.weight * dnorm(z, mean= z.obs[i], sd= tau) # particle weight based on true observations
(z.obs)
p.weight <- p.weight/sum(p.weight) # Normalizing the weights
xest <- sum(x.update * p.weight) # Final particle filtering estimation is calculated as expected value of
the states using their weights
xest.out <- cbind(xest.out, xest) # Save data in a vector for later plotting
p.weight.before <- p.weight # Keep the particle weights before resampling
#----resampling-----
idx <- sample(1:N, size = N, replace = T, prob = p.weight) # randomly sample index numbers 1:N with
replacemnt
and based on the particle weights
x <- x.update[idx] # select particles with higher weights based on
generated indexes
p.weight <- rep(1/N, N) # Assign equal weights to all particles again
#-----
state.mat <- cbind(state.mat, x) # save all the particles states at each time step
w.mat <- cbind(w.mat, p.weight.before)
}#
op <- par(family = "serif") #Seleccionar el font
#plot(timestep, x.true, pch= 20, cex=0.8, col= 10, xlab= "Timestep" , ylab = "Deficiencia", xlim=c(0,
Lfinal),panel.first=grid())
#for (ix in (1:N)){
# lines(timestep[1:L], state.mat[ix,], col="grey") # plotting particle's trajectories
#}
#points(timestep, x.true, pch= 20, cex=0.8, col= 10)
#points(timestep[1:L],xest.out[1:L], pch="*", col= "green", cex= 0.8) # plotting final PF estimation
#legend( "topleft", legend= c("Deficiencia", "Particle Filtering estimations", "Particles trajectories"), pch=
c(19,
20, NA), lty= c(NA, NA, 1), col= c("red", "green", "grey82"), cex=0.5)
#title("Particle Filtering Estimation of Pump1 deficiency")
## ***** ##
## ***** Prognosis ***** ##
## ***** ##
## now we have a set of particles at last timestep with their weights. Suppose there is no more
observation available
## We keep the weights constant and only propagate the particles in time using the process model
for (ii in (L+1):(Lfinal)){
dx <- (31960425259582555*((20*ii)/9021 - 5210/3007)^3)/162507888954036977664 - (1183*((20*ii)/9021
-
5210/3007)^2)/9396875 - (83819194824768725*ii)/91623979140897973469184 +
49085763944500680994349/38176657975374155612160000
ruido <- exp(rnorm(N, mean = 0, sd = sigma))

```

```

rand <- (floor(runif(N, 0, 2)) * 1)
ruido <- ruido*rand
x.prognos <- x + dx*delta.t*ruido
x <- x.prognos # Assign new generated states to
the particles
state.mat <- cbind(state.mat, x)
xest <- sum(x * p.weight.before)
xest.out <- cbind(xest.out, xest) # Save data in arrays for later
plotting
}
#----- Plotting the prognostics particles
plot(timestep, x.true, pch= 20, cex=0.8, col= 10, xlab= "Timestep" , ylab = "Deficiencia", xlim=c(0,
Lfinal),panel.first=grid(),ylim=c(0,1.5))
for (iii in (1:N)){
lines(timestep[1:L], state.mat[iii, 1:L], col="grey") # plot particles in monitoring period again
lines(timestep[L:Lfinal], state.mat[iii,L:Lfinal], col="lightsteelblue") #Particulas pronostico
}p
oints(timestep[1:1953], x.true, pch= ".", col= 10, cex= 4, xlab= "Timestep" , ylab = "Eficiencia") #x true rojo
points(timestep[1:L],xest.out[1:L], pch="*", col= 11, cex= 0.8)
points(timestep[L:Lfinal],xest.out[L:Lfinal], pch="*", col= 12, cex= 0.8) #Deficiencia pronostico
legend("topleft", legend=c("Eficiencia", "Estimacion FP", "Prediccion Eficiencia FP"),col=c("red", "green",
"blue"),
pch=c(20,20,20),cex=0.8)
title("Predicción Filtro de Partículas")
#Estimacion MTTF
treshold = numeric()
for (hora in (L+1):(Lfinal)){ #hora es el instante de tiempo
n <- sum(state.mat[1:N,hora] >= 1 & state.mat[1:N,hora] <= 1.01)
#n <- sum(state.mat[1:N,hora] >= 1)
#print(n)
treshold <- cbind(treshold, n)
}c
ol <- ncol(treshold)
treshold <- treshold[1:col]
timestep2 <- seq(L+1, by= delta.t, length.out= col)
#Suma Ponderada.
suma <- 0
horas <- 0
for (indice in 1:length(treshold)){
valor <- treshold[indice]*timestep2[indice]
suma <- suma + valor
horas <- horas + timestep2[indice]
}M
TTF <- suma/sum(treshold)
print('MTTF')
print(MTTF)
print('x.true')
print(length(x.true))
plot(timestep2,treshold)
lines(timestep2,treshold)
abline(v= MTTF, lty=3, col= 12)
abline(v=length(x.true),lty=3, col = 10)

```