ARTÍCULOS

BOLETÍN CIENTÍFICO TECNOLÓGICO

ACADEMIA POLITÉCNICA MILITAR

MANTENIMIENTO PREDICTIVO UTILIZANDO MATLAB

SR. JOSÉ PEREDA BARRALES

MANTENIMIENTO PREDICTIVO UTILIZANDO MATLAB

Sr. José Pereda Barrales.[1]

Resumen: *El mantenimiento predictivo hace referencia a la supervisión inteligente del equipamiento a fin de evitar fallos futuros. Al contrario que el mantenimiento preventivo convencional, el programa de mantenimiento no está determinado por un cronograma prescrito; en su lugar, se establece mediante algoritmos analíticos que utilizan los datos recopilados por los sensores de los equipos. Los algoritmos son cruciales para el éxito del mantenimiento predictivo. El procesamiento previo de los datos de sensores se realiza mediante técnicas estadísticas y de procesamiento de señales avanzadas. En el presente caso, se emplean técnicas de aprendizaje automático (Machine Learning) para calcular el estado del equipamiento. Mediante el uso del software Matlab, se analiza el caso del censado de variables de un motor de avión Turbofan, que fue expuesto en el seminario de mantenimiento predictivo dictado por el ingeniero Gerardo Hernández Correa (Ingeniero de aplicación de Mathworks) el día 15 de noviembre de 2017, en el Hotel Marriot.*

Palabras claves: *Mantenimiento predictivo, aprendizaje automatizado, matlab, inteligencia artificial, análisis de componentes principales.*

Abstract: *Predictive maintenance refers to the intelligent supervision of the equipment in order to avoid future failures. Unlike conventional preventive maintenance, the maintenance program is not determined by a prescribed schedule; instead, it is established by analytical algorithms that use the data collected by the equipment sensors.*

Algorithms are crucial to the success of predictive maintenance. The pre-processing of the sensor data is made by statistical techniques and advanced signal processing. In the present case, automatic learning techniques (Machine Learning) are used to calculate the state of the equipment. By using Matlab software, the measured variables of a Turbofan engine's case is analyzed, which was exposed in a predictive maintenance seminar, by the engineer Gerardo Hernández Correa (Ma-

1 Ingeniero Civil en Electrónica, Universidad Iberoamericana de Ciencias y Tecnoloía. Magíster en Currìculum y Evaluación, Universidad de Aconcagua.

nager application engineer) on November 15th, 2017, at Marriot Hotel.
Keywords: *Predictive maintenance,machine learning, matlab, artificial intelligence, principal component analisis.*

1. INTRODUCCIÓN

Existen diversos tipos de mantenimiento, los que se dividen en prefalla y posfalla entre ellos el preventivo, el programado y el predictivo. El primero es aquel que se enfoca en la reparación de averías, roturas, fallas o cualquier inconveniente que afecte el desempeño o las funciones de una cosa u objeto, suele realizarse en casos en los que se descomponen herramientas y aparatos, por ejemplo; en las fábricas en las que la maquinaria sufre alguna avería o desperfecto, que impide su funcionamiento normal e, incluso, dejan de funcionar por completo, entonces se realizan las reparaciones, reposiciones de piezas y demás movimientos necesarios para que la máquina u objeto, vuelva a desempeñar su trabajo correctamente. El mantenimiento preventivo programado está enfocado en la prevención de daños. Este tipo suele estar programado, para realizarse cada cierto tiempo, independientemente de que existan o no muestras de deterioro en equipos y objetos. Por otro lado, el predictivo es un tipo de mantenimiento preventivo a partir del que se realizan las intervenciones para realizar arreglos, se atienen a un seguimiento y vigilancia del funcionamiento y del rendimiento, se determina su evolución, con lo que se anticipa, de esta manera, en qué momento deben realizarse las reparaciones y ajustes correspondientes.

El mantenimiento predictivo ofrece las siguientes ventajas para los clientes y los fabricantes de equipamiento: reducción del tiempo de inactividad de los equipos gracias a la identificación de los problemas antes de que se produzcan fallos, lo que permite una planificación idónea de las tareas de revisión y el aumento de la vida útil del equipamiento; determinación automática de la causa raíz del fallo, lo que permite la realización de las reparaciones apropiadas sin necesidad de destinar recursos a establecer un diagnóstico; ahorro de costos por tareas de mantenimiento innecesarias.

Una alternativa para realizar este seguimiento del estado de la maquinaria es mediante sensores dispuestos en puntos claves para poder medir variables relevantes del estado de funcionamiento de la máquina. Una vez tomados los datos, se procesan con un computador y, así se detectan anomalías de forma prematura. La técnica de procesamiento de datos que se analiza es el *Machine Learning* o aprendizaje automatizado.

Una manera fácil de procesar los algoritmos de *Machine Learning* es con el *software Matlab*, el que tiene incluido dentro de sus herramientas específicas algoritmos estadísticos y de regresión que son fácilmente ejecutados mediante opciones preestablecidas para este tipo de cálculos. Además, se pueden procesar grandes cantidades de datos.

2. DESARROLLO. MACHINE LEARNING

La herramienta de inteligencia artificial, *Machine Learning*, ocupa datos con los que produce un programa y, de esa manera, realiza una tarea específica. Esta técnica ocupa una serie de algoritmos para identificar patrones en los datos y, así, poder apoyar en la toma de decisiones.

Existen dos tipos de aprendizaje automatizado: el aprendizaje supervisado y el no supervisado. El primero necesita datos de entrada y de salida para poder desarrollar un modelo predictivo y, con esto, desarrollar algoritmos de clasificación o de regresión. Por otro lado, el aprendizaje no supervisado, utiliza únicamente datos de entrada, con lo que crea algoritmos de clasificación o agrupamiento de datos *(Clustering)* (Del Pozo, 2016).

En general, la forma de trabajo del *Machine Learning* se organiza en dos grandes etapas, independientemente de su forma de aprendizaje. La primera es de entrenamiento, la que tiene, a su vez, diferentes subetapas, que considera en primera instancia la carga de los datos; luego, el ajuste de los datos para que estos puedan ser procesados por el computador; posteriormente, se aplican los algoritmos internos de aprendizaje para finalizar con la creación de un modelo, el que se aplicará en la segunda etapa.

La segunda etapa es llamada de predicción, la que tiene las siguientes subetapas: primero, se entran nuevos datos para poder aplicar lo aprendido y poder aplicar el modelo creado en la etapa anterior; luego, se aplica la misma etapa de ajuste de datos, para después aplicarlos al modelo creado y, finalmente, obtener una predicción.

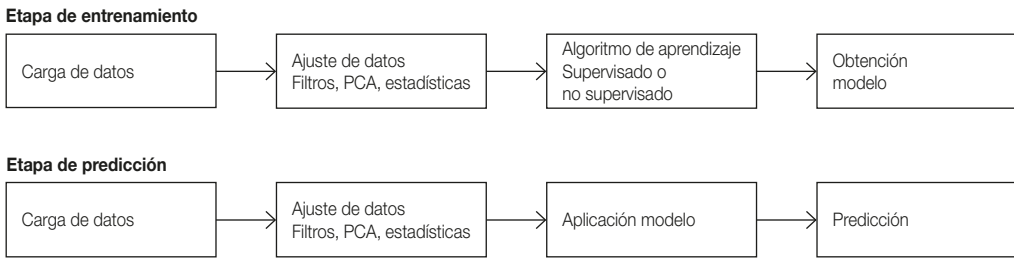


Figura Nº 1:"Etapas de trabajo del *Machine Learning*".

Fuente: Mathworks Inc., Hernández (2017).

El método de aprendizaje se basa en la utilización de algoritmos estadísticos que son capaces de generalizar comportamientos y reconocer patrones a partir de una información suministrada anteriormente en forma de ejemplos. El objetivo es predecir

las posibles situaciones futuras habiendo aprendido anteriormente de las situaciones pasadas, es decir, aprender de la experiencia.

2.1. Dificultades y características del Machine Learning

El *Machine Learning* se utiliza comúnmente cuando hay muchas variables a analizar y el sistema es muy complejo para poder determinar una ecuación o plantear una hipótesis o simplemente identificar alguna tendencia o dependencia. Por tanto, el computador utiliza algoritmos de aprendizaje basados en funciones de costo, como la de mínimos cuadrados, por ejemplo:

$$J = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2$$

Donde:
m, es el número de datos de entrenamiento.

h $\Theta(x^{i})$, es la salida de cada ensayo i que predice nuestra función de hipótesis.

y i es la salida exacta de cada ensayo i.

El objetivo es minimizar la función de costos J(Θ), para que después se pueda aplicar los valores de ajustados de Θ para aplicar el modelo de salida h(Θ).

Un tipo de modelo de salida o función de hipótesis es el modelo de regresión lineal que tiene la siguiente expresión:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Donde el valor n es el número de características de entrada.

Las dificultades que presentan el modelo anteriormente descrito suelen ser las siguientes:

- La diversidad de los datos, ya que muchos se presentan en diversos formatos y magnitudes, los cuales deben ser ajustados y normalizados, para que los algoritmos los puedan procesar.
- El procesamiento de los datos previo al entrenamiento implica el filtrado, selección y transformación de los datos de variables más influyentes en el sistema, ya que un modelo muy complejo es muy difícil que pueda aprender.

- El entrenamiento o aprendizaje del modelo suele ser un proceso iterativo y de prueba y error hasta que se obtenga el modelo que cumpla con los requerimientos establecidos.
- Calidad o rendimiento del modelo. Existe un compromiso entre velocidad, precisión y complejidad del modelo. No es posible obtener un modelo con todas esas características óptimas.

3. EL SOFTWARE MATLAB

Matlab es un entorno de cálculo técnico de altas prestaciones para cálculo numérico y visualización. Integra: análisis numérico, cálculo matricial, procesamiento de señales, gráficos, entre otros. Es un entorno fácil de usar, donde los problemas y las soluciones son expresados como se escriben matemáticamente, sin la programación tradicional. El nombre *Matlab* proviene de "MATrix LABoratory'" (Laboratorio de Matrices). *Matlab* fue escrito originalmente para proporcionar un acceso sencillo al *software* matricial desarrollado por los proyectos LINPACK[2] y EISPACK,[3] que juntos representan lo más avanzado en programas de cálculo matricial. *Matlab* es un sistema interactivo cuyo elemento básico de datos es una matriz que no requiere dimensionamiento. Esto permite resolver muchos problemas numéricos en una fracción del tiempo que llevaría hacerlo en lenguajes como C, BASIC o FORTRAN. *Matlab* ha evolucionado en los últimos años a partir de la colaboración de muchos usuarios. En entornos universitarios se ha convertido en la herramienta de enseñanza estándar para cursos de introducción en álgebra lineal aplicada, así como cursos avanzados en otras áreas. En la industria, *Matlab* se utiliza para investigación y para resolver problemas prácticos de ingeniería y matemáticas, con un gran énfasis en aplicaciones de control y procesamiento de señales. *Matlab* también proporciona una serie de soluciones específicas denominadas *TOOLBOXES*.

4. APLICACIÓN DE UN CASO DE USO: MOTOR DE AVIÓN TURBOFAN

La explicación del caso de uso es de un motor de avión Turbofan que se expuso en el seminario en Santiago de Chile el día 15 de noviembre, llamado "Mantenimiento predictivo usando *Matlab*". En este se realizó un análisis de los datos de 14 sensores correspondientes a 100 motores de avión Turbofan, los que fueron entregados a los asistentes, más el material de presentación, el que, complementado con información recabada en el sitio web de *Matlab*, se explica la metodología de *Machine Learning* aplicada al mantenimiento predictivo.

---

2 Software desarrollado por Argone National Laboratory, para cálculos científicos.

3 Conjunto de subrutinas en fortran, para el cálculo de autovalores y autovectores, desarrollado por Argone National Laboratory.

Figura Nº 2: “Motor Turbofan”.

Fuente: Mathworks Inc., Hernández (2017).

4.1. Antecedentes del monitoreo de datos

Los datos de los sensores son temperatura, velocidades de rotación, presiones, entre otras, que completan 14 sensores, de 100 motores del mismo tipo Turbofan, a lo largo de 125 vuelos.

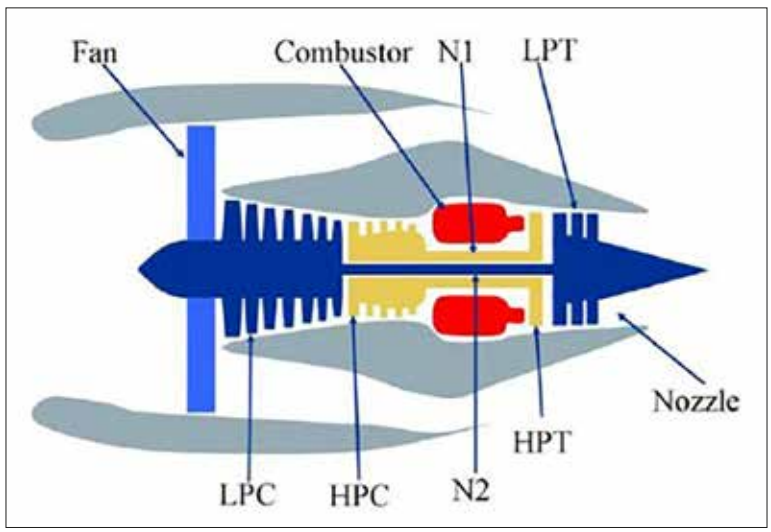


Figura Nº 2: “Motor Turbofan”.

Fuente: Mathworks Inc., Hernández (2017).

Con los datos provenientes de los sensores se requiere realizar un mantenimiento predictivo, el que pretende lo siguiente:

- Detectar fallas.
- Predecir el tiempo para el mantenimiento.
- Identificar componentes defectuosos.
- Incrementar la disponibilidad.
- Reducir los costos de mantenimiento.

El procedimiento general para realizar este mantenimiento es el siguiente:

- Entrenar un modelo para predecir cuándo las fallas ocurrirán.
- Desplegar información de los sensores.
- Predecir fallas en tiempo real.
- Importar y analizar datos históricos de los sensores.

4.2 Importar y analizar datos históricos de los sensores

Los datos de los sensores se pueden originar de dos formas o dos escenarios posibles, lo que incide en la estrategia de aprendizaje.

4.2.1 Escenario 1: no existen datos de fallas

En este escenario únicamente existen datos de los sensores, pero no están disponibles los datos de falla, los que son en definitiva los datos de salida. Lo anterior implica que, al contar con únicamente datos de entrada, solo se puede optar un aprendizaje no supervisado y que el modelo que se obtenga no serviría para realizar predicciones, únicamente agrupaciones de datos o *clustering.* Este es el esquema de datos que se trabajará en el presente trabajo.

4.2.2 Escenario 2: existen datos de fallas

En este escenario existen datos de sensores y, además, datos de fallas ocurridas, por tanto, hay datos de entrada y de salida. Con tal información se puede ejecutar un aprendizaje supervisado, en el cual es posible obtener un modelo que sea capaz de realizar predicciones y clasificaciones.

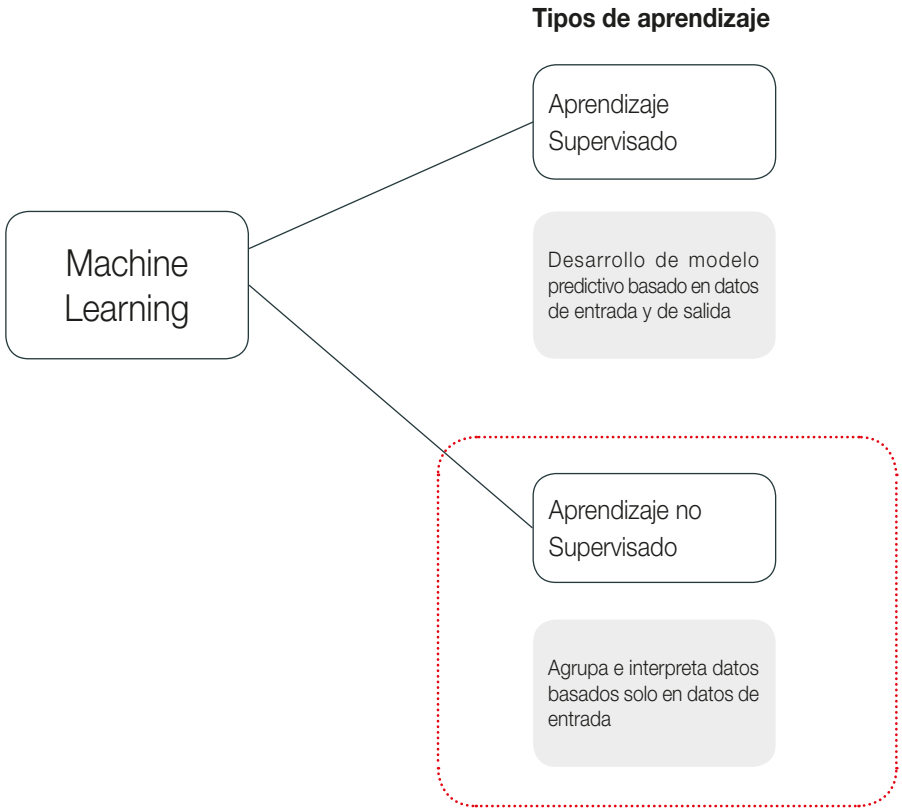


Figura Nº 3: "Tipos de aprendizaje del Machine Learning".

Fuente: Mathworks Inc., Hernández (2017).

En la figura anterior se indica que se aplica para el aprendizaje una técnica de aprendizaje no supervisada, debido a que no hay datos de fallas disponibles (datos de salida) y solamente se tienen los datos de los sensores, que corresponden a datos de entrada. Por lo tanto, el modelo podrá generar agrupación de datos o *Clustering*.

4.3. Procedimiento de Machine Learning para el modelo de mantenimiento predictivo

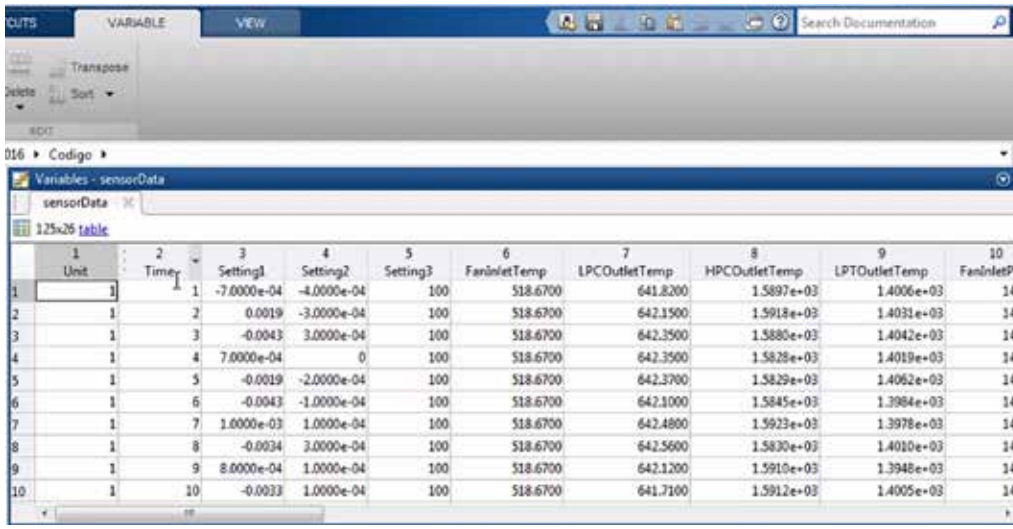
En esta parte se realizarán todos los pasos para poder obtener el modelo de mantenimiento predictivo, hasta llegar al modelo final y las pruebas con el modelo obtenido.

4.3.1 Paso 1: lectura de datos de los sensores

Lo primero es leer los datos de los sensores, para lo cual se ejecuta el siguiente comando:

```
% read in data, assuming here that one data file can fit into memory

sensorData = readtable('train_FD001_Unit_1.csv','ReadVariableNames',true);
```



	1 Unit	2 Time	3 Setting1	4 Setting2	5 Setting3	6 FanInletTemp	7 LPCOutletTemp	8 HPCOutletTemp	9 LPTOutletTemp	10 FanInletPr
1	1	1	-7.0000e-04	-4.0000e-04	100	518.6700	641.8200	1.5897e+03	1.4006e+03	14
2	1	2	0.0019	-3.0000e-04	100	518.6700	642.1500	1.5918e+03	1.4031e+03	14
3	1	3	-0.0043	3.0000e-04	100	518.6700	642.3500	1.5880e+03	1.4042e+03	14
4	1	4	7.0000e-04	0	100	518.6700	642.3500	1.5828e+03	1.4019e+03	14
5	1	5	-0.0019	-2.0000e-04	100	518.6700	642.3700	1.5829e+03	1.4062e+03	14
6	1	6	-0.0043	-1.0000e-04	100	518.6700	642.1000	1.5845e+03	1.3984e+03	14
7	1	7	1.0000e-03	1.0000e-04	100	518.6700	642.4800	1.5923e+03	1.3978e+03	14
8	1	8	-0.0034	3.0000e-04	100	518.6700	642.5600	1.5830e+03	1.4010e+03	14
9	1	9	8.0000e-04	1.0000e-04	100	518.6700	642.1200	1.5910e+03	1.3948e+03	14
10	1	10	-0.0033	1.0000e-04	100	518.6700	641.7100	1.5912e+03	1.4005e+03	14

Figura Nº 4: "Vista parcial de los datos de los sensores".

Fuente: Mathworks Inc., Hernández (2017).

4.3.2. Paso 2: visualización de los datos

Una vez cargados los datos, se procede a hacer una selección de los 14 sensores y mostrar una visualización gráfica de los datos obtenidos. Para realizar lo anterior, se ejecutan los siguientes comandos:

```
% Interactive visualizations to aid discovery
figure
for ii = 1:9
subplot(3,3,ii)
plot(sensorData.Time,sensorData{:,5+ii})
title(sensorData.Properties.VariableNames{5+ii})
xlabel('Time')
xlim([0,125])
end
%% select relevant variable names based on visualization
variableNames={'Unit''TotalHPCOutletPress''FuelFlowRatio''CorrFanSpeed'…
'CorrCoreSpeer''BypassRatio''BleedEnthalpy''BleedEnthalpy''HPTCoolantBleed'…
'LPTCoolantBleed'}
```

Al ejecutar las líneas anteriores, resulta el siguiente gráfico que es una muestra de 9 sensores:

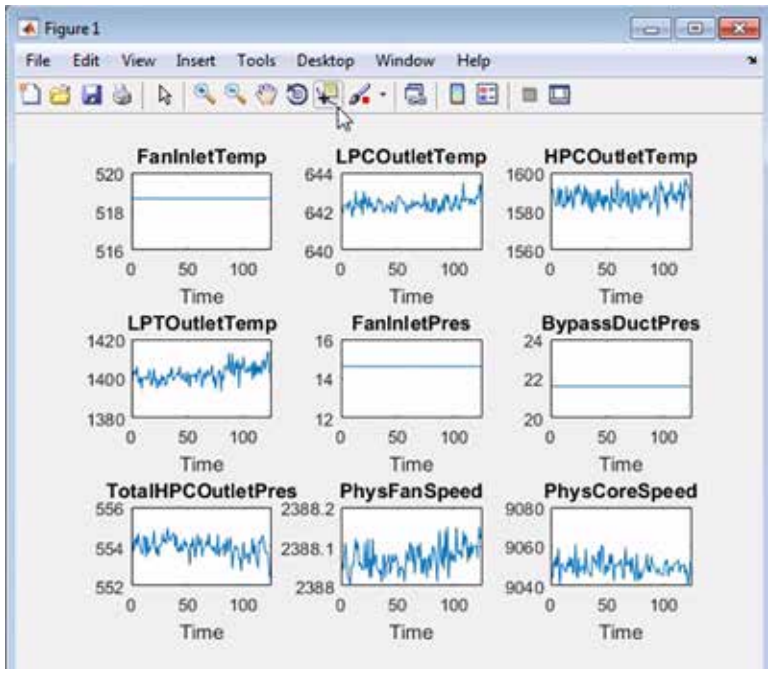


Figura Nº 5: Gráfico de los nueve sensores seleccionados

Fuente: Mathworks Inc., Hernández (2017).

Como se puede apreciar, no es posible identificar ninguna tendencia o adelantar alguna conclusión con respecto al mantenimiento.

4.3.3. Filtrado de datos

Como generalmente los datos de los sensores vienen con ruido, es conveniente y necesario aplicarles filtros digitales para poder obtener información más pura de los sensores. Para ello, se aplica el siguiente código:

```
% Remove noise

% Filter configuration


filterWindow=5;
b=(1/filterWindow)*ones(1,filterWondow);
a=1;
% filter sensor data
smoothData=sensorData;
smoothData{:,3:end}=filter(b,a,sensorData{:,3:end});
smoothData{1:5,:}=[];
```

El anterior es un filtro de media móvil que sirve para suavizar los datos de los sensores más importantes, que difieren de los sensores de la figura 5, por tratarse de sensores filtrados.

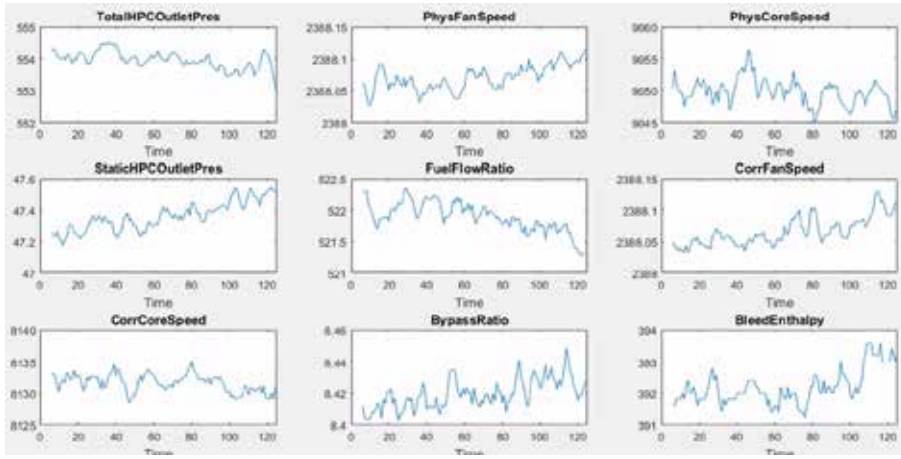


Figura N° 6: "Gráfico de los datos filtrados".

Fuente: Mathworks Inc., Hernández (2017).

Dentro de este mismo punto de visualización, se puede obtener otro tipo de gráfico que el de control, en donde se puede observar la variación de los datos y si algún dato se sale del rango de desviación estándar. Lo anterior se materializa con el siguiente comando:

```
% control chart
controlchart(sansorData.LPCOutletTemp,'chart','i')
```

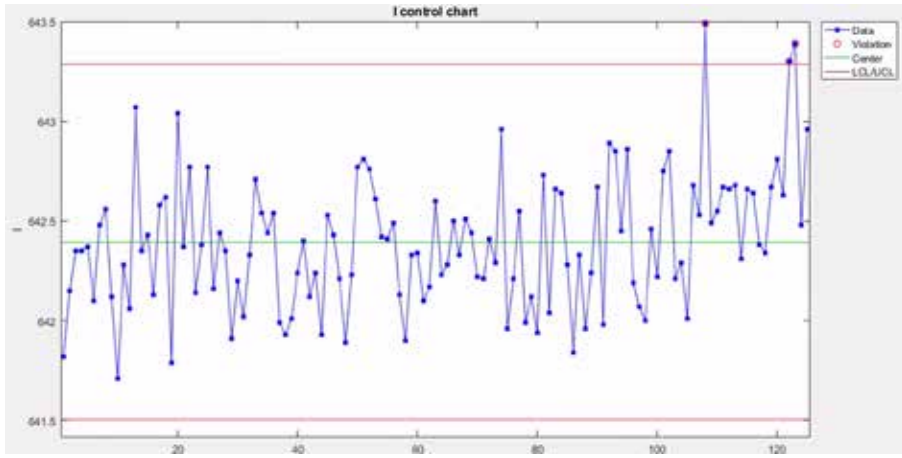


Figura Nº 7: "Gráfico de control de un sensor".

Fuente: Mathworks Inc., Hernández (2017).

El gráfico muestra una línea del centro que corresponde a la media y la línea de los extremos es la desviación típica, los puntos que sobrepasan los límites de la línea roja, podrían estar representando un problema.

Dado que el anterior es solo el gráfico de un solo sensor, esta información podría ser suficiente para poder tomar una decisión, pero como son muchos sensores, el problema debe ser tratado con una herramienta más poderosa como el *Machine Learning* y procesar la información de todos los sensores a la vez.

4.3.4. Análisis de componentes principales

El análisis de componentes principales (PCA) es una técnica estadística para poder determinar en cuántos componentes se puede representar la mayor variabilidad de los datos y, de esa forma, trabajar con menos gráficos, pero con toda la información proyectada (la más relevante), en los ejes de los componentes principales.

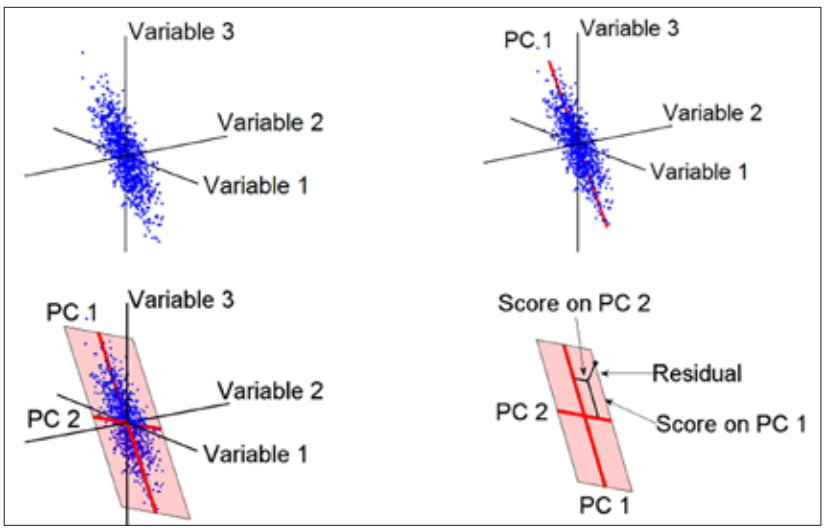


Figura Nº 8: "Ejemplo de los componentes principales".

Fuente: Mathworks Inc., Hernández (2017).

En la figura 8 se busca representar en pocas dimensiones la mayor cantidad de información, en ejes ortogonales llamados componentes principales, los cuales representan la mayor desviación de los valores en el espacio.

Para poder realizar el análisis de componentes principales se deben estandarizar los datos para que estos tengan media cero y varianza 1, ya que si se mezclan cantidades tales como rpm, temperaturas, no se podrían trabajar apropiadamente

dada las diferentes magnitudes que poseen. Para lo anterior se utiliza el siguiente procedimiento, en cuya parte final hay un comando que ejecuta el análisis de componentes principales.

```
% Standardize data
Xtrain=smoothDataAll{:,3:end};
% Give mean of zero and standar desviation of one
XtrainMean=mean(Xtrain);
XtrainStd=std(Xtrain);
XtrainStandard=(Xtrain-repart(XtrainMean,length(Xtrain),1))./...
repmat(XtrainStd,length(Xtrain),1);
[coeff,score,latent]=pca(XtrainStandard);
```

El siguiente gráfico representa el resultado del análisis PCA:

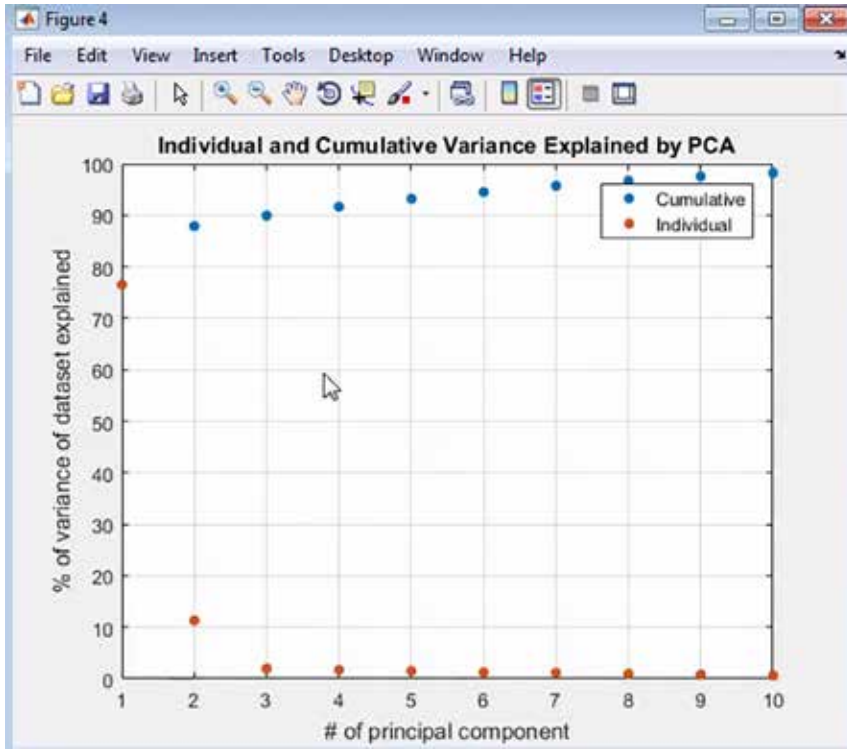


Figura Nº 9: "Gráfico de varianza por componente principal".

Fuente: Mathworks Inc., Hernández (2017).

Se puede apreciar en el gráfico anterior que bastan solo 2 componentes principales, la primera representa el 76% de la varianza y la segunda representa más del 10% de variación de los datos. A partir del tercer componente principal ya la variación baja a menos del 2%. Por lo que se toman solo 2 componentes principales para trabajar con

toda la información de los sensores. En otras palabras, de un espacio de 14 dimensiones (una por sensor) se lleva la información a un espacio de solo 2 dimensiones que representan la proyección de las 14.

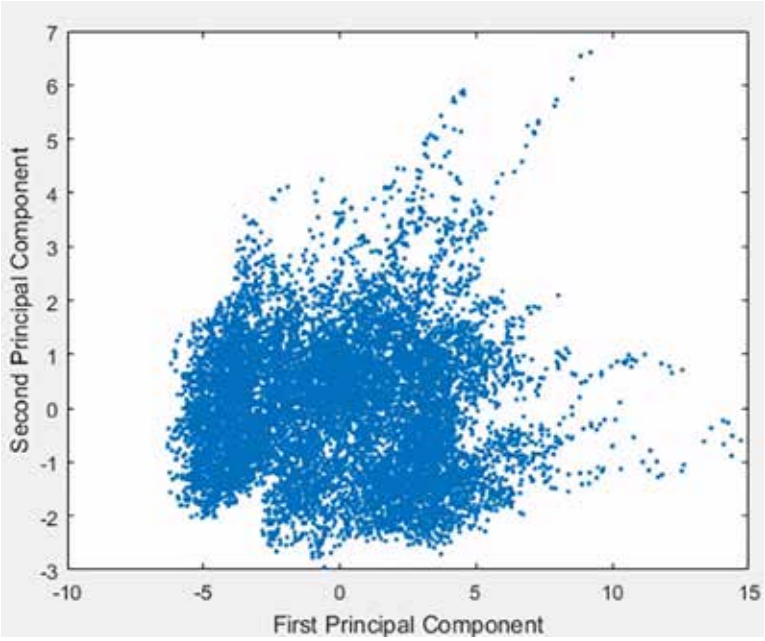


Figura Nº 10: "Gráfico de los dos componentes principales".

Fuente: Mathworks Inc., Hernández (2017).

En la figura N° 10 se logra apreciar la dispersión de los valores proyectados en los dos componentes principales, en donde se puede apreciar la mayor varianza de los datos. De esa forma, es más fácil procesar un algoritmo de aprendizaje, ya que se tienen solo dos variables.

4.3.5. Visualización de información más detallada de los componentes principales

Es muy útil para comenzar los análisis del gráfico de la figura N° 10 la visualización de todos los datos. Así, se analizan los centroides y cómo evolucionan los datos desde la primera muestra hasta la última. Para esto se ejecuta el siguiente código:

```
% Find early and late data
earlyData=score(smoothDataAll.Time <= 100,1:2);
lateData=score(smoothDataAll.Time >100 ,1:2);

% compute centroids
```

```
EarlyCent=[mean(earlyData(:,1)),mean(earlyData(:,2))];
LateCent=[mean(lateData(:,1)),mean(LateData(:,2))];

figure;hold on
plot(earlyData(:,1),earlyData(:,2),'.')
plot(lateData(:,1),lateData(:,2),'x.')
scatter(EarlyCent(1),EarlyCent(2),50,'g','^','filled')
scatter(LateCent(1),LateCent(2),50,'g','o','filled')
holdoff
legend('Early Points','Late Points','Early Centroids','Late
Centroid','Location','Northwest')
xlabel('First Principal Component')
ylabel('Second Principal Component')
title('Early (Sample<100) and Late (Sample>=100) for Each Engine')
```

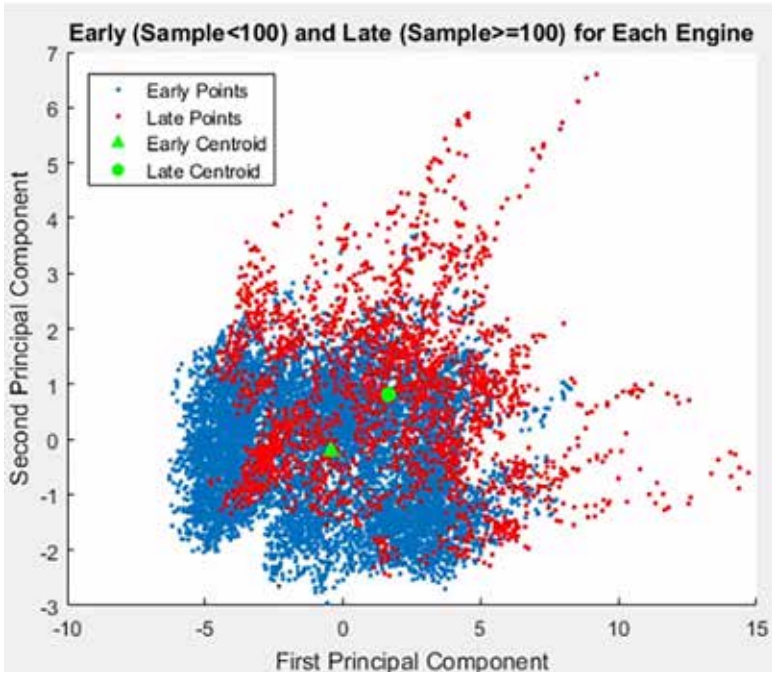


Figura Nº 11: "Gráfico de los dos componentes principales antes y después del vuelo 100".

Fuente: Mathworks Inc., Hernández (2017).

Se logra apreciar los valores antes y después del vuelo 100; en azul (punto más oscuro) es antes del vuelo 100 y en verde están los datos después del vuelo 100, lo que indica que los valores tienden a alejarse del centro y refleja que los motores se alejan de la parte central del dibujo, que serían las condiciones normales. A medida que los puntos se alejan del centro (en rojo o punto menos oscuro), tienden a salirse del rango de normalidad y a necesitar mantenimiento.

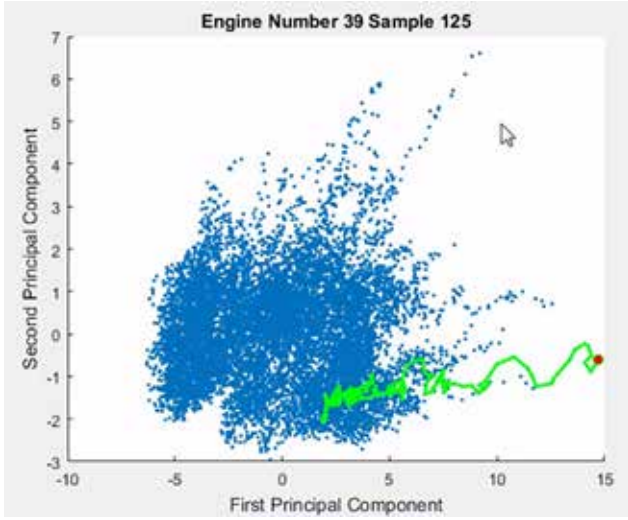


Figura Nº 12: "Evolución de un motor tras las diversas tomas de datos".

Fuente: Mathworks Inc., Hernández (2017).

En el gráfico anterior, se puede observar cómo un motor se va alejando de la condición de "normalidad" a medida que se van realizando las diversas medidas.

4.3.6. Resultado Final: los rangos para el mantenimiento

Finalmente, después de correr el algoritmo de *Machine Learning* se logran determinar los rangos para clasificar los motores en el grado de urgencia para realizar el mantenimiento. Para lo cual, se ejecutó el siguiente código:

```
idxAlarm=score(:,1)>10 | score(:,)<-7 | score(:,2)>5 | score(:,2)<-4;
idxWarm=score(:,1)>5 | score(:,)<-6.5 | score(:,2)>2 | score(:,2)<-3 ...
& ~idxAlarm;
figure;hold on
patch([-10;-10;15;15;10;10;-7;-7;-10],...
    [-4,8,8,-4,-4,5,5,-4,-4],'r','FaceAlpha',0,3)
patch([-7,-7,10,10,5,5,-6.5,-6.5,4.99,4.99,-6.5],...
    [-4,5,5,-4,-4,2,2,-3,-3,-4,-4],'y','FaceColor',[1 .8 0],'FaceAlpha',0.3)
patch([-6.5,-6.5,5,5,-6.5],[-3,2,2,-3,-3],'g','FaceAlpha',0.3)
plot(score(:,1),score(:,2),'.')
xlabel('First principal component')
ylabel('Second principal component')
```

De esta forma, se definieron tres zonas representadas con tres colores:

- Color verde: zona normal, requiere mantenimiento a largo plazo.
- Color amarillo: zona de advertencia, requiere mantenimiento en un mediano plazo.
- Color rojo: zona de alarma, requiere mantenimiento de inmediato.

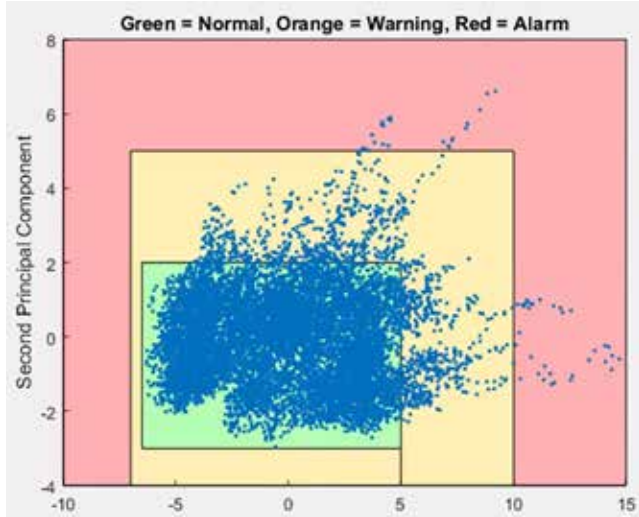


Figura Nº 13: "Zonas para el mantenimiento".

Fuente: Mathworks Inc., Hernández (2017).

El *software* indica, además, que el 90% de los datos están en régimen normal.

```
>>runPhmMonitor(10)

Percent ofpointscapturedby'normal'conditions:90.0%
Percent ofpointscapturedby'normal'+'warm'conditions:99.5%
```

Al realizar un gráfico de la evaluación de los vuelos, en la medida que van fallando, se tienen los siguientes resultados:

Figura Nº 14: Gráfico de porcentaje de motores, según la clasificación de mantenimiento.

Fuente: Mathworks Inc., Hernández (2017).

En el gráfico se muestra la evolución de los motores, según las cantidades de vuelos hasta que fallen. Se puede apreciar que, de la totalidad de motores en condición normal (verde) a partir de los 160 vuelos aproximadamente, comienzan a reducir su cantidad y a desplazarse a la condición de alerta (naranja) y, a medida que aumentan los vuelos, siguen decreciendo hasta comenzar a estar en estado de alarma (rojo). Si aumenta aún más la cantidad de vuelos, el 100% de los motores falla y no hay motores ni en condición normal o de alerta.

5 CONCLUSIONES

El *software Matlab* permite, mediante varias opciones, graficar datos de diferentes formas para apoyar la toma de decisiones. En otro sentido, facilita el procesamiento de grandes cantidades de datos de forma muy fácil y amigable para el usuario.

Es mejor contar con datos de salida para el *Machine Learning*, dado que esto permite realizar predicciones de cuando fallarían los motores, lo que no se puede lograr solamente con los datos de entrada.

La técnica no supervisada de aprendizaje no permite realizar predicciones, solamente determinar qué equipos o partes requieren mantenimiento de manera urgente, medianamente urgente o si no se necesita mantener por el momento, ya que la técnica no supervisada permite monitorear el estado del equipo o de algún componente a lo largo de su uso, lo cual solo da indicios de la variación de su estado.

Por tanto, sería posible, al perfeccionar las técnicas de mantenimiento por *Matlab*, programar indicadores de mantenimiento como por ejemplo, confiabilidad, disponibilidad, etcétera.

6. BIBLIOGRAFÍA

[1] DEL POZO GALLEGO, Carlos. “Aplicación de técnicas de Machine Learning con regularización al diagnóstico de fallos en motores de inducción”. Director: Óscar Duque Pérez [Trabajo de fin de grado]. Universidad de Valladolid, España,2016.

[2] HERNÁNDEZ CORREA,Gerardo. *Seminario de mantenimiento predictivo usando Matlab*. Santiago de Chile: Hotel Marriot. 15 de noviembre de 2017.

[3] MATHWORKS INC.[en línea] [Fecha de consulta: 15 de noviembre de 2017]. Disponible en: https://www.mathworks.com/

[4] PASCUAL, Rodrigo (2005). *El arte de mantener*. Santiago de Chile: Ediciones Universidad de Chile.