

Detección de fallas en vehículos aéreos no tripulados mediante señales de orientación y técnicas de aprendizaje de máquina

López-Estrada, F.R.^a, Méndez-López, A.^a, Santos-Ruiz, I.^a, Valencia-Palomo, G.^{b,*}, Escobar-Gómez, E.^a

^aTecnológico Nacional de México / I.T. Tuxtla Gutiérrez, TURIX-Dynamics Diagnosis and Control Group, Carretera Panam. km 1080, CP 29050, Chiapas, México.
^bTecnológico Nacional de México / I.T. Hermosillo, Av. Tecnológico y Periférico Poniente, S/N, 83170, Sonora, México.

To cite this article: López-Estrada, F.R., Méndez-López, A., Santos-Ruiz, I., Valencia-Palomo, G., Escobar-Gómez, E., 2021. Fault detection in unmanned aerial vehicles via orientation signals and machine learning. Revista Iberoamericana de Automática e Informática Industrial 18 254-264. <https://doi.org/10.4995/riai.2021.14031>

Resumen

Este trabajo propone un esquema de detección y localización de fallas en los actuadores de un vehículo aéreo no tripulado (VANT) del tipo cuadrirrotor. Para ello, se considera un enfoque basado en datos haciendo uso de técnicas de aprendizaje de máquina. En este enfoque se construye un modelo implícito del sistema a través de la información proporcionada por los sensores del VANT. Primero, a través de un plataforma de vuelo de tipo giroscópica, se captan las vibraciones correspondientes a la orientación, posición angular y aceleración lineal cuando el vehículo se encuentra en vuelo estacionario en condiciones nominales. Estos datos se procesan mediante Análisis en Componentes Principales (PCA) para la extracción de características. Posteriormente, se induce una falla a los actuadores a través de un recorte en cada una de las hélices del VANT que ocasionan una reducción del empuje generado por los rotores. Estos datos se proyectan también al subespacio de componentes principales y se comparan con los datos nominales. Para discernir entre los datos nominales y los datos cuando el vehículo presenta falla, se emplea el estadístico T^2 de Hotelling. Finalmente, el desarrollo se complementa con los algoritmos de clasificación de k -vecinos más cercanos (k -NN) y de máquina de vectores de soporte (SVM). Los resultados muestran una tasa de clasificación correcta del 89.6 % (k -NN) y 92.4 % (SVM) respectivamente para 423 conjuntos de datos de validación.

Palabras clave: Vehículo aéreo no tripulado, detección e identificación de fallas, análisis en componentes principales, aprendizaje de máquina, cuadrirrotor.

Fault detection in unmanned aerial vehicles via orientation signals and machine learning

Abstract

This work proposes an actuator fault detection and isolation scheme for a quadrotor unmanned aerial vehicle (UAV) under a data-driven approach using machine learning techniques. In this approach, an implicit model of the system is built through the information provided by the onboard sensors of the UAV. First, using a tailored flying platform, vibrations corresponding to the orientation, angular position and linear acceleration were captured with the UAV flying in hover mode under nominal conditions. This data is processed by Principal Component Analysis (PCA) for feature extraction. Subsequently, faults in the actuators are induced through a cut in each of the UAV propellers which generate a reduction in the thrust of the rotors. These data are also projected into the PCA subspace and compared to the nominal data. Hotelling's T^2 statistic is used to discern between nominal data and data when the vehicle exhibits an actuator fault. Finally, the developed algorithms were complemented with k -nearest neighbors (k -NN) and support vector machine (SVM) classification algorithms. The results show a correct classification rate of 89.6 % (k -NN) and 92.4 % (SVM) respectively for 423 validation datasets.

Keywords: Unmanned aerial vehicle, fault detection and isolation, principal component analysis, machine learning, quadrotor.

*Autor para correspondencia: gvalencia@hermosillo.tecnm.mx

1. Introducción

Un vehículo aéreo no tripulado (VANT) de tipo cuadrirrotor es una aeronave sin piloto humano que se opera a través de una entrada electrónica iniciada por un controlador de vuelo o un sistema de control de vuelo autónomo a bordo (Nonami et al., 2010). Este tipo de vehículos tiene un creciente interés debido a que proveen una gran área de aplicaciones en sectores como: transporte, exploración del medio y agricultura (Sharifi et al., 2010), vigilancia, búsqueda y rescate, seguridad, militar (Mouloua et al., 2001), entre otros. En VANTs, los accidentes ocurren con frecuencia debido a fallas en los sistemas electrónicos, rotores y sensores (Mueller and D'Andrea, 2014; Guzmán-Rabasa et al., 2019) lo que ocasiona daño parcial o total al vehículo generando pérdidas económicas o daños a terceros. Debido a ello, es de gran importancia la detección y localización de las fallas en el vehículo para mejorar su seguridad y confiabilidad ante comportamientos anómalos (López-Estrada et al., 2019, 2020).

Existen dos metodologías para detectar fallas durante el vuelo del VANT; las basadas en modelos matemáticos y las basadas en datos. Las primeras se basan en leyes físicas que rigen el movimiento traslacional y rotacional del VANT. Se generan señales residuales que comparan mediciones de sensores con estimaciones obtenidas del modelo (Saied et al., 2015; Guzmán-Rabasa et al., 2019), donde las discrepancias se interpretan como una posible falla. Aunque este enfoque ha logrado aplicarse de manera exitosa (Vey and Lunze, 2016; Xian and Hao, 2019), usualmente es difícil parametrizar el modelo del sistema, debido a que algunos parámetros como los momentos de inercia, empujes y constantes aerodinámicas, entre otros, no son fáciles de medir o estimar. En contraparte, los métodos basados en datos utilizan un modelo implícito a partir de los datos empíricos del sistema caracterizando una zona de operación nominal y zonas de operación con fallas (Gertler, 2015; Guo et al., 2019). Con ello, se realiza una comparación entre los datos nominales y con fallas para identificar anomalías en el vehículo.

Algunos trabajos que consideran métodos basados en datos se describen a continuación. En Yap (2014) se presentó el monitoreo de salud estructural de VANTs usando una unidad de medición inercial (IMU) unida al eje de un cuadrirrotor para analizar la vibración característica mediante la transformada rápida de Fourier. En Ghalamchi and Mueller (2018) se propuso la detección de fallas analizando el espectro de vibración en las trayectorias de vuelo mediante la instalación de un acelerómetro en el centro del cuerpo del cuadrirrotor. El argumento de esta técnica, se basa en que las hélices dañadas aplican vibración adicional al sistema y la detección de esas fallas ayuda a prevenir daños en otros componentes. Los autores en Li et al. (2016) propusieron un análisis en componentes principales (PCA) dinámico caracterizando zonas de operación nominal y con fallas. Los resultados obtenidos demuestran la efectividad de los métodos basados en PCA, sin embargo solo se realizaron pruebas en simulación sin una validación experimental.

Otros autores proponen la combinación de técnicas basadas en datos y técnicas de aprendizaje de máquina con la finalidad de generar métodos de diagnóstico más eficientes. Por ejemplo, Jiang et al. (2015) consideraron una metodología basada en redes neuronales artificiales y ondeletas para procesar las señales de vibración producto de la aceleración en el vehículo.

También se combinaron métodos acústicos y redes neuronales artificiales, como en Iannace et al. (2019), donde a partir de las mediciones del ruido producido por la rotación de la hélice, se construyó un modelo de clasificación para detectar una hélice desequilibrada. Algunos trabajos consideran un enfoque combinando sistemas basados en datos y sistemas basados en modelo, por ejemplo, en Freeman et al. (2013) se consideró la generación residual basada en modelo y la detección de anomalías basada en datos para un VANT y los aplicó a los datos experimentales de pruebas de vuelo con y sin fallas. Este enfoque, se aplicó también a la detección de fallas en sensores, e.g. en Sun et al. (2017) se presenta un sistema de inferencia neuro-difuso adaptativo (ANFIS) que combina un mecanismo de entrenamiento de datos en línea con un sistema de decisión basado en ANFIS.

En trabajos más recientes, los autores en Benini et al. (2019) proponen un sistema de clasificación de fallas para un VANT basado en análisis frecuencial y la extracción de características, logrando aplicarlo al sistema experimental Songbird. En Wang et al. (2019) se propone un sistema de detección de anomalías en sensores basado en redes neuronales recurrentes de corto periodo de tiempo. El sistema se aplicó satisfactoriamente usando datos de sensores de orientación. Otro trabajo de detección de fallas considerando redes neuronales recurrente se propuso en Xiao et al. (2019), donde se logró detectar interferencias en el sistema GPS que afectan la navegación del vehículo. En Yang et al. (2020) se considera un enfoque híbrido para fallas en sensores basados en el análisis secuencial de residuos y máquinas de vectores de soporte. No obstante, a pesar de que el tema se ha investigado ampliamente, el problema no está del todo resuelto y sigue siendo de amplio interés para la comunidad científica. En particular el diagnóstico de fallas en actuadores. Este problema es de vital importancia, pues debido al fuerte acoplamiento de los mismos, una falla en un actuador afecta al desempeño del vehículo en las dinámicas de orientación y traslación.

Por otro lado, una de las desventajas de los métodos basados en datos es que para sistemas de dinámicas rápidas, se genera una gran cantidad de datos, lo cual reduce la aplicabilidad de los algoritmos de detección de fallas. Una forma de afrontar este problema es a través de la extracción de la información relevante de los datos, y por ende la reducción de su dimensión. Esta reducción se puede lograr mediante técnicas de análisis de componentes principales como se demostró recientemente en el trabajo de Wang et al. (2020), en donde aplicando técnicas de PCA se logró procesar datos masivos, con algoritmos computacionalmente eficientes, en la detección de fallas en un VANT de ala fija.

El diagnóstico de fallas en VANTs fue abordado recientemente mediante técnicas de aprendizaje automático que usan clasificadores. Por ejemplo, Baskaya et al. (2017) usaron clasificadores basados en máquinas de vectores de soporte (SVM), Yousefi et al. (2018) propusieron un modelo de detección basado en el análisis discriminante lineal, y Alos and Dahrouj (2020) propusieron un clasificador basado en los k -vecinos más cercanos (k -NN). El común denominador en estos sistemas de detección de fallas mediante aprendizaje automático es la dificultad para distinguir entre diferentes tipos de falla, por lo que es importante una selección adecuada de las variables de entrada (*features*) del sistema, de modo que en el espacio definido

por esas variables se maximice la separabilidad de las fallas. Por ello, se asume que el preprocesamiento mediante PCA de los datos de vuelo del VANT (la entrada del clasificador) puede contribuir a mejorar esta separabilidad.

En este trabajo se considera una técnica basada en PCA para la detección de fallas en los actuadores del VANT y se complementa con algoritmos de clasificación k -NN y SVM para la identificación de la falla. Para ello se genera inicialmente un modelo implícito del sistema a partir de datos obtenidos de sensores proporcionados por la tarjeta de control de vuelo. El criterio de selección subyacente es que el diagnóstico de fallas basado en datos permite evitar la carga del modelado preciso requerido en el enfoque basado en un modelo matemático. El análisis se realiza a partir de datos de la orientación, velocidad angular y aceleración lineal del vehículo obtenidos de la IMU inmersa en la tarjeta de control de vuelo del VANT. El conocimiento de las características del sistema en condiciones de operación libre de fallas es útil para plantear escenarios de detección y localización de fallas como un problema de tratamiento digital de señales. El trabajo se plantea en términos de clasificación de datos a partir de la extracción de sus características principales en condiciones nominales y de la comparación con los datos del vehículo cuando se induce una falla. Las pruebas experimentales se realizaron sobre una plataforma de vuelo de tipo giroscopio y los resultados demuestran la efectividad del método propuesto.

Este artículo está organizado de la siguiente manera: en la Sección 2, se presenta el análisis en componentes principales y su uso en la detección de fallas; en la Sección 3, se presentan los algoritmos de clasificación considerados; la Sección 4 describe la aplicación del método propuesto a la detección de fallas del VANT; la Sección 5 presenta el análisis de los resultados; finalmente, la Sección 6 presenta las conclusiones del trabajo.

2. Detección de fallas basado en el análisis en componentes principales

El análisis en componentes principales (PCA por sus siglas en inglés, *Principal Component Analysis*) es una técnica estadística multivariada que se usa ampliamente en el monitoreo de procesos industriales (Russell et al., 2012). Mediante cálculos matriciales básicos el PCA permite construir un modelo basado en datos (*data-driven*) del comportamiento de un proceso, el cual puede ser utilizado después para determinar si las mediciones actuales de las variables del proceso se ajustan estadísticamente al comportamiento nominal (dentro de ciertos límites de confianza) o si se infiere la falla u otro comportamiento anómalo.

El análisis de componentes principales se usa ampliamente en el monitoreo de plantas complejas con cientos de variables porque, al revelar relaciones lineales entre las variables, reduce significativamente la dimensionalidad del modelo de planta. Esta técnica es un enfoque estadístico, que permite la transformación de datos multivariados a un espacio de menor dimensión sin pérdida de generalidad (Jolliffe, 2011). Los nuevos componentes lineales son combinaciones lineales de las variables originales, con la característica adicional de que ahora se presentan como independientes. El PCA consta de dos fases: entrenamiento y monitoreo. En la fase de entrenamiento se crea

un modelo implícito de planta a partir de los datos empíricos. En la fase de monitoreo, este modelo se utiliza para la FDI.

Para llevar a cabo el proceso del PCA se requiere de construir la matriz de mediciones

$$\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_n] \in \mathbb{R}^{m \times n}; \quad (1)$$

de tal forma que se construye la matriz \mathbf{X} como:

$$\mathbf{X} = \begin{bmatrix} x_1(1) & x_2(1) & \dots & x_n(1) \\ x_1(2) & x_2(2) & \dots & x_n(2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(m) & x_2(m) & \dots & x_n(m) \end{bmatrix}, \quad (2)$$

donde n representa el número de variables de proceso y m es el número de muestras tomadas para cada variable.

Posteriormente se requiere escalar los vectores de datos de \mathbf{X} de tal forma que se tenga una media cero y varianza unitaria, para luego formar la matriz de puntuaciones estándar $\mathbf{Z} = [\mathbf{z}_1 \quad \mathbf{z}_2 \quad \dots \quad \mathbf{z}_n] \in \mathbb{R}^{m \times n}$ con

$$\mathbf{z}_k = \frac{1}{\sigma_k} (\mathbf{x}_k - \mu_k \mathbf{1}_m), \quad k = 1, 2, \dots, n; \quad (3)$$

donde $\mathbf{1}_m = [1 \quad 1 \quad \dots \quad 1]^T \in \mathbb{R}^m$, μ_k y σ_k son la media aritmética y la desviación estándar de \mathbf{x}_k , respectivamente:

$$\mu_k = \frac{1}{m} \sum_{i=1}^m x_k(i), \quad \sigma_k = \sqrt{\frac{1}{m} \sum_{i=1}^m (x_k(i) - \mu_k)^2}. \quad (4)$$

La estandarización en (3) permite una ponderación equitativa de la variabilidad de los datos debido a que las variables de proceso tienen sus valores en diferentes rangos y unidades de medida. A partir de esta matriz de datos se inicia la extracción de características del sistema. Esta técnica consiste en encontrar una matriz cuadrada ortogonal, que se define como:

$$\mathbf{P} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \dots \quad \mathbf{p}_n] \in \mathbb{R}^{n \times n}; \quad (5)$$

donde \mathbf{P} debe expresar el dato en $\mathbf{Z} \in \mathbb{R}^{m \times n}$ en términos de una nueva matriz con las nuevas coordenadas de datos $\mathbf{T} = [\mathbf{t}_1 \quad \mathbf{t}_2 \quad \dots \quad \mathbf{t}_n] \in \mathbb{R}^{m \times n}$ referidas a la base ortonormal \mathbf{p}_k :

$$\mathbf{T} = \mathbf{Z}\mathbf{P}. \quad (6)$$

Esto se ilustra en la Figura 1, donde el vector unitario \mathbf{p}_1 (primera componente) sigue la dirección de máxima varianza de los datos, mientras que \mathbf{p}_2 (segunda componente) representa una varianza residual, en la dirección perpendicular a \mathbf{p}_1 .

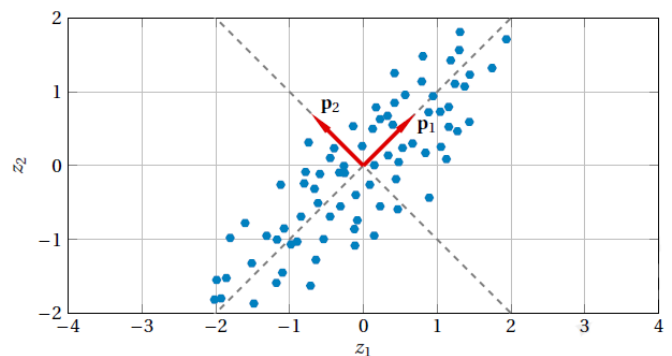


Figura 1: Base ortonormal para el cambio de coordenadas en el PCA, caso bidimensional.

Las mediciones estandarizadas pueden recuperarse a partir de \mathbf{T} y \mathbf{P} :

$$\mathbf{Z} = \mathbf{TP}^T. \quad (7)$$

Las columnas \mathbf{t}_n de la matriz \mathbf{T} son las puntuaciones correspondientes a la nueva variable, sin significado físico, las cuales son expresadas como combinaciones lineales de las variables originales \mathbf{z}_k agrupadas en \mathbf{Z} . Una forma de obtener la base \mathbf{P} es a través de una descomposición en valores singulares (SVD) de \mathbf{Z} (Strang et al., 2016). Otra forma es mediante una descomposición en eigenvalores (EVD) de la matriz de covarianza de \mathbf{Z} . La EVD se computa en menos tiempo que la SVD cuando el número de observaciones m , supera el número de variables n , pero es menos precisa. La matriz de covarianza se calcula mediante la siguiente ecuación:

$$\mathbf{S} = \text{cov}(\mathbf{Z}) = \frac{1}{m-1} \mathbf{Z}^T \mathbf{Z}. \quad (8)$$

Posteriormente, se calcula una EVD de \mathbf{S} , con lo que se obtiene la expresión factorizada de la matriz de covarianza:

$$\mathbf{S} = \mathbf{PAP}^T, \quad (9)$$

donde $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ es una matriz diagonal con sus eigenvalores en orden decreciente:

$$\mathbf{\Lambda} = \text{diag}([\lambda_1 \lambda_2 \dots \lambda_n]), \quad \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0, \quad (10)$$

mientras que \mathbf{P} es la matriz formada con los eigenvectores ortonormales, donde \mathbf{p}_k es el eigenvector correspondiente a λ_k . Puede demostrarse que λ_k es igual a la varianza de la k -ésima componente principal (Jolliffe, 2011):

$$\lambda_k = \text{var}(\mathbf{t}_k) = \frac{1}{m-1} \mathbf{t}_k^T \mathbf{t}_k, \quad (11)$$

de modo que la fracción de la varianza total explicada por las primeras q componentes principales está dada por:

$$\text{TEV}(q) = \frac{\sum_{k=1}^q \lambda_k}{\sum_{k=1}^n \lambda_k}, \quad (12)$$

cuyo denominador es n cuando se usan datos normalizados con varianza unitaria.

Si una λ_k es cero, su componente principal correspondiente es descartable, pues no contribuye en nada para explicar la variabilidad de los datos. Incluso se pueden descartar las componentes con valores pequeños distintos de cero. El número de componentes principales que sí son significativas para explicar la variabilidad de los datos puede determinarse por validación cruzada (Wold, 1978) y por otros métodos (Tamura and Tsujita, 2007), o bien fijando de manera arbitraria el porcentaje de la varianza que las componentes principales deban explicar. Cuando se determina el número de componentes principales significativas q , es posible expresar que $\mathbf{P} = [\bar{\mathbf{P}} \tilde{\mathbf{P}}]$ donde $\bar{\mathbf{P}} = [\mathbf{p}_1 \mathbf{p}_2 \dots \mathbf{p}_q] \in \mathbb{R}^{n \times q}$, denominada *matriz de carga*, genera el *subespacio principal* que contiene la variabilidad esencial del proceso, mientras que $\tilde{\mathbf{P}} = [\mathbf{p}_{q+1} \mathbf{p}_{q+2} \dots \mathbf{p}_n] \in \mathbb{R}^{n \times (n-q)}$ genera el *subespacio residual* asociado al ruido y a las variaciones débilmente correlacionadas con la física del proceso. Análogamente, es posible hacer una separación en la matriz de las nuevas coordenadas, $\mathbf{T} = [\bar{\mathbf{T}} \tilde{\mathbf{T}}]$ donde $\bar{\mathbf{T}}$ contiene las coordenadas

de las proyecciones de los datos en el subespacio principal y $\tilde{\mathbf{T}}$ contiene las proyecciones del subespacio residual. Puesto que los subespacios principal y residual son ortogonales, la información que contiene uno no la contiene el otro, de manera que se complementan:

$$\mathbf{Z} = [\bar{\mathbf{T}} \tilde{\mathbf{T}}][\bar{\mathbf{P}} \tilde{\mathbf{P}}]^T = \bar{\mathbf{T}}\bar{\mathbf{P}}^T + \tilde{\mathbf{T}}\tilde{\mathbf{P}}^T = \bar{\mathbf{Z}} + \tilde{\mathbf{Z}}. \quad (13)$$

Puede considerarse que las matrices \mathbf{P} y $\mathbf{\Lambda}$ junto con los vectores de medias $\boldsymbol{\mu} = [\mu_1 \mu_2 \dots \mu_n]$ y que las varianzas $\boldsymbol{\sigma} = [\sigma_1 \sigma_2 \dots \sigma_n]$ usadas en la estandarización constituyen un modelo basado en datos del proceso. A continuación, se describe cómo utilizar este modelo para detectar fallas.

Con el modelo PCA definido por $\{\mathbf{P}, \mathbf{\Lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma}\}$ es posible inferir si cualquier nueva medición $\mathbf{x}_{\text{new}} = [x_1 \ x_2 \ \dots \ x_n]^T$ es compatible con el comportamiento nominal del proceso. Para ello se estandariza \mathbf{x}_{new} con las medias y desviaciones estándar del modelo. Luego, la muestra estandarizada \mathbf{z}_{new} puede ser expresada en términos de la base ortonormal \mathbf{P} con las coordenadas:

$$\mathbf{t}_{\text{new}} = \mathbf{z}_{\text{new}}^T \mathbf{P}. \quad (14)$$

Las proyecciones de la muestra \mathbf{z}_{new} sobre los subespacios principal y residual están dadas por:

$$\bar{\mathbf{z}}_{\text{new}} = \bar{\mathbf{P}}\bar{\mathbf{P}}^T \mathbf{z}_{\text{new}}, \quad (15)$$

$$\tilde{\mathbf{z}}_{\text{new}} = \tilde{\mathbf{P}}\tilde{\mathbf{P}}^T \mathbf{z}_{\text{new}} = (\mathbf{I} - \bar{\mathbf{P}}\bar{\mathbf{P}}^T) \mathbf{z}_{\text{new}}, \quad (16)$$

de modo que $\mathbf{z}_{\text{new}} = \bar{\mathbf{z}}_{\text{new}} + \tilde{\mathbf{z}}_{\text{new}}$. Para determinar la existencia de fallas no es necesario comparar la muestra actual \mathbf{z}_{new} con todos los datos de entrenamiento en \mathbf{Z} (ni con su versión transformada \mathbf{T}), porque pueden utilizarse estadísticos de prueba. En este caso se usa el T^2 de Hotelling

$$T^2(\mathbf{z}_{\text{new}}) = \sum_{k=1}^q \frac{t_k^2}{\lambda_k} = \mathbf{t}_q^T \mathbf{\Lambda}_q \mathbf{t}_q = \mathbf{z}_{\text{new}}^T \bar{\mathbf{P}} \mathbf{\Lambda}_q^{-1} \bar{\mathbf{P}}^T \mathbf{z}_{\text{new}}, \quad (17)$$

donde $\mathbf{\Lambda}_q = \text{diag}([\lambda_1 \lambda_2 \dots \lambda_q])$ contiene sólo los eigenvalores asociados al subespacio principal. El argumento \mathbf{z}_{new} en (17) indica que T^2 se calcula para la muestra de prueba, aunque también es posible calcularlos para cada renglón de la matriz de entrenamiento \mathbf{Z} , pero estos pueden ser calculados mediante una distribución de probabilidad sin necesidad de un cálculo exhaustivo. Cuando el proceso opera en condiciones nominales, el estadístico T^2 tiene valores pequeños y se infiere la existencia de fallas cuando sobrepasan el umbral U_{T^2} (Figura 2).

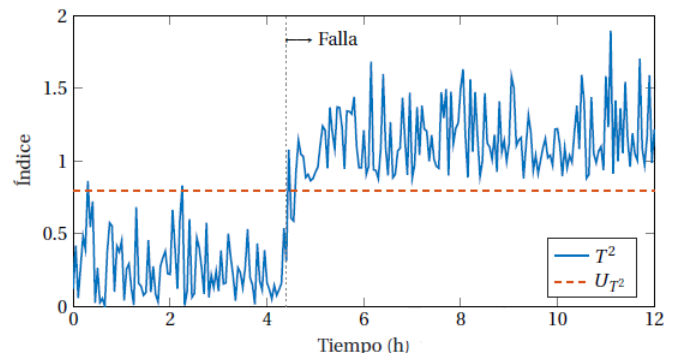


Figura 2: Monitoreo de procesos mediante el índice T^2 . Se infiere una falla cuando $T^2 > U_{T^2}$, aunque el umbral se sobrepasa ocasionalmente durante la operación normal.

En Russell et al. (2012), se establece U_{T^2} como:

$$U_{T^2} = \frac{q(m^2 - 1)}{m(m - q)} F_{\alpha}(q, m - q), \quad (18)$$

donde $F_{\alpha}(q, m - q)$ representa el percentil α en la distribución F de Fisher con q y $m - q$ grados de libertad. Generalmente, $0.95 \leq \alpha \leq 0.99$, según la confianza y sensibilidad deseadas. Si el número de muestras de entrenamiento m es grande puede usarse la aproximación de Qin (2012):

$$U_{T^2} \approx \chi_{\alpha}^2(q), \quad (19)$$

que corresponde a la distribución chi cuadrada de Pearson con nivel de confianza α y q grados de libertad. En términos geométricos, considerando el caso bidimensional, la región definida por la desigualdad $T^2(\mathbf{z}) \leq U_{T^2}$ es una elipse que contiene, con cierto nivel de confianza, a los puntos compatibles con la operación nominal del proceso. Mientras que en el exterior de la elipse se ubican los puntos asociados con alguna condición de falla. Se puede verificar que la región de confianza para T^2 tiene geometría elipsoidal partiendo de la definición de T^2 en (17). Así, en el caso bidimensional, la región de confianza en el plano (t_1, t_2) es una elipse, porque la frontera de dicha región (donde $T^2 = U_{T^2}$) queda definida por la ecuación $t_1^2/\lambda_1 + t_2^2/\lambda_2 = U_{T^2}$.

3. Localización de fallas mediante clasificadores

El PCA en conjunto con el estadístico de T^2 de Hotelling presentado en la sección anterior aporta información sobre la ocurrencia de una falla, sin embargo, no es posible determinar su ubicación, por lo que es necesario utilizar métodos de clasificación. En esta sección se presentan dos algoritmos de clasificación investigados para la identificación de la falla.

3.1. Algoritmo de clasificación k -NN

El algoritmo k -NN (del inglés *k-Nearest Neighbors*) es una técnica de aprendizaje automático que clasifica muestras correspondientes a múltiples clases comparando las similitudes (ceranía) entre la muestra de prueba y las diferentes muestras de un conjunto de datos de entrenamiento. Las muestras se conforman por mediciones de las variables relevantes del problema, las cuales son denominadas características o atributos (*features*, en inglés). Además de los atributos, cada muestra del conjunto de entrenamiento contiene una etiqueta que identifica la clase a la que pertenece, de modo que el aprendizaje k -NN es de tipo supervisado.

En el entrenamiento, el clasificador k -NN no realiza ningún cálculo y se limita a almacenar las muestras etiquetadas, difiriendo todo el cálculo para la etapa de inferencia donde pretende identificar la clase correspondiente a muestras de prueba no etiquetadas (Santos-Ruiz et al., 2019). Es un método no paramétrico, pues no resume los datos disponibles en un conjunto de parámetros, por lo que todas las muestras de entrenamiento y sus etiquetas deben permanecer en memoria durante su periodo de operación. Para la inferencia, asume que una muestra dada pertenece a la clase mayoritaria entre sus k vecinos más cercanos. Para esto, deben definirse dos cosas: el número de vecinos a utilizar y cómo medir la ceranía entre muestras.

Formalmente, una muestra de entrenamiento es un vector $\mathbf{x} \in \mathbb{R}^n$, donde cada componente x_i es un atributo de la muestra (i.e., una variable de orientación del vehículo). Por cada muestra de entrenamiento se asigna una etiqueta $y(\mathbf{x})$, indicando la clase a la que corresponde (i.e., el tipo de falla en el vehículo). Con esa base de conocimiento, el clasificador k -NN pretende determinar la clase correspondiente a muestras de prueba no etiquetadas. Se representa por $\hat{y}(\mathbf{x}^*)$ a la etiqueta de clase asignada a una muestra de prueba \mathbf{x}^* por el clasificador.

En el marco de la teoría de decisión bayesiana, la clasificación de una muestra de prueba consiste en hallar, dentro de q posibles clases, aquella con la mayor probabilidad *a posteriori* $P(c_j | \mathbf{x}^*)$, para $j = 1, \dots, q$, donde \mathbf{x}^* es el vector de características de la muestra y c_j representa la j -ésima clase. Mediante el teorema de Bayes, puede demostrarse (Martínez and Martínez, 2015) que la clase con la máxima probabilidad *a posteriori*, y por ende la salida del clasificador, está dada por

$$\hat{y}(\mathbf{x}^*) = \arg \max_j P(c_j | \mathbf{x}^*) = \arg \max_j \frac{k_j}{k}, \quad (20)$$

donde k_j es el número de muestras de entrenamiento que pertenecen a la j -ésima clase, considerando sólo los k vecinos más cercanos a la muestra de prueba. En resumen, la salida del algoritmo k -NN es la etiqueta con la mayor frecuencia relativa entre los k vecinos más cercanos a \mathbf{x}^* .

La ceranía entre muestras se determina a partir de la distancia entre los puntos que las representan en el espacio de las características. Bajo la métrica euclidiana, la distancia entre dos muestras, \mathbf{x} y \mathbf{x}' , está dada por:

$$d(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}')^T (\mathbf{x} - \mathbf{x}')}. \quad (21)$$

Geoméricamente, la región que encierra a las k muestras de entrenamiento más cercanas a la muestra de prueba es una hipersfera, una circunferencia en el caso bidimensional. En este sentido, el algoritmo k -NN consiste en calcular la densidad de puntos correspondiente a cada clase en la hipersfera y luego seleccionar la clase con la mayor densidad.

La elección del número k de vecinos se hace minimizando el error de clasificación en el conjunto de muestras de prueba, el cual se mide mediante el índice de pérdidas:

$$\text{loss} = 1 - \frac{m_s}{m}, \quad (22)$$

donde m es el número total de muestras en el conjunto de prueba, y m_s es el número de estas muestras que resultan correctamente clasificadas para un valor dado de k . Para mejorar la eficiencia del clasificador, en la determinación del valor óptimo de k se usa validación cruzada iterativa (*K-fold cross-validation*), de modo que el error de clasificación se calcula para un conjunto de muestras distinto al de entrenamiento. Por lo general, los valores pequeños de k producen clasificadores muy sensibles al ruido de medición.

3.2. Algoritmo de clasificación SVM

La máquina de vectores de soporte (SVM, del inglés *Support Vector Machine*) es otro algoritmo de clasificación basado en el aprendizaje supervisado. A diferencia del k -NN, donde sólo interesa minimizar el error de clasificación, en la SVM existe otro objetivo de optimización: maximizar el *margen* de clasificación. El margen es la distancia entre el hiperplano que

separa las clases (la superficie de decisión) y las muestras de entrenamiento más cercanas a ese hiperplano, las cuales se denominan *vectores de soporte* (Vapnik, 2013).

En el caso de la clasificación binaria (dos clases, identificadas como $y = \{-1, 1\}$), el entrenamiento de la SVM consiste en resolver un problema de optimización cuadrática que hallará el hiperplano óptimo para clasificar las características de entrada $\mathbf{x}^{(i)}$ en dos clases:

$$\text{Minimizar: } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi^{(i)}, \quad \text{para } i = 1, 2, \dots, N, \quad (23)$$

$$\text{sujeto a: } w_0 + \mathbf{w}^\top \mathbf{x}^{(i)} \geq 1 - \xi^{(i)}, \quad \text{si } y^{(i)} = 1, \quad (24)$$

$$w_0 + \mathbf{w}^\top \mathbf{x}^{(i)} \leq -1 + \xi^{(i)}, \quad \text{si } y^{(i)} = -1, \quad (25)$$

donde N es el número de muestras de entrenamiento y \mathbf{w} contiene los coeficientes del hiperplano de separación. Las $\xi^{(i)}$ son *variables flexibles* que relajan las restricciones de separabilidad para permitir la convergencia de la optimización cuando los datos de entrenamiento no son linealmente separables. La variable C controla la penalización por errores de clasificación.

Para construir la superficie de decisión sólo se requieren los vectores de soporte seleccionados de los datos de entrenamiento. Una vez entrenada la SVM, el resto de los datos de entrenamiento son irrelevantes. En las clasificaciones no binarias (multiclase), como ocurre en el reconocimiento de múltiples fallas a partir de datos obtenidos por sensores, puede utilizarse una estrategia de clasificación *uno contra todos*. En estos casos, se construyen múltiples clasificadores binarios, uno por cada clase; en cada uno de ellos se considera que la clase correspondiente es positiva ($y = 1$), mientras que el resto pertenece a una clase complementaria negativa ($y = -1$). Esta es la estrategia utilizada en el presente trabajo para clasificar las fallas por pérdida de empuje según el rotor donde ocurren.

4. Aplicación a la detección e identificación de fallas en actuadores al VANT

En esta sección se describen los experimentos llevados a cabo para la generación de resultados. En la primera parte se caracteriza el actuador del VANT y, en la segunda parte, se inducen las fallas al vehículo para probar el algoritmo de detección y localización de fallas.

4.1. Caracterización de la pérdida de empuje

El VANT considerado en este trabajo está construido en un marco en forma de cruz de 450 mm de distancia entre rotor y rotor. La tarjeta de vuelo es una ErleBrain 3, la cual combina una computadora Linux integrada (Raspberry Pi 2) que ejecuta ArduPilot, cuenta con soporte para el Sistema Operativo Robotico (ROS) y una tarjeta secundaria que integra puertos de entrada/salida, electrónica de potencia y varios sensores como una unidad de medición inercial, brújula, sensor barométrico de altura y un GPS externo. Los actuadores del VANT son cuatro motores sin escobillas Emax 2212 en conjunto con hélices de 10×4.5 pulgadas.

Dado que la falla más común en este tipo de vehículos es la pérdida total o parcial de empuje, se realizó la caracterización de sus actuadores (motores). Para ello se realizaron experimentos que simulan una pérdida gradual de empuje: con la hélice

completa y con una hélice recortada 1 cm y 2 cm. Esta reducción de longitud se aprecia en la Figura 3. Para caracterizar la pérdida de empuje al recortar la hélice se diseñó un banco experimental como se observa en la Figura 4. Para la medición de la velocidad del rotor fue necesario conectar un sensor óptico a una tarjeta de adquisición de datos NIDAQ USB-6002 que contabilizaba pulsos. En el extremo del banco, se colocó una báscula digital de precisión para cuantificar la fuerza de empuje. Para todos los experimentos, la velocidad del rotor se ajustó a un valor fijo de aproximadamente el 80 % de su velocidad máxima lo cual se logró fijando el ciclo de trabajo del PWM que alimenta el control de velocidad del rotor con una tasa de muestreo de 50 Hz.

El experimento se realizó tres veces para cada escenario para garantizar la calidad de los resultados. Como se aprecia en la Tabla 1, al disminuir la dimensión de la hélice, disminuye la fuerza de empuje. Esto puede traducirse como una pérdida de eficiencia en el rotor, con lo que reduciendo el tamaño de la hélice es posible simular una falla en el actuador del vehículo. Por ejemplo, con la hélice completa se tiene un empuje máximo de 38.53 N, mientras que recortándola 1 cm se tiene un empuje de 30.04 N lo que se traduce en una pérdida de 22 %. Esta pérdida de empuje en un sistema de vuelo debe ser compensado por los tres rotores restantes y como resultado genera vibraciones anómalas que pueden caracterizarse para la identificación del rotor fallado.

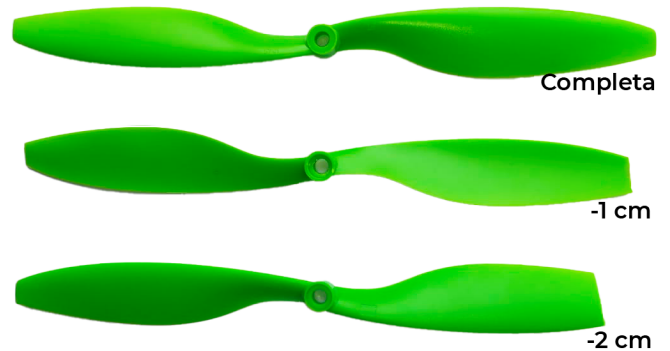


Figura 3: Recorte en las hélices para simulación de pérdida de empuje.

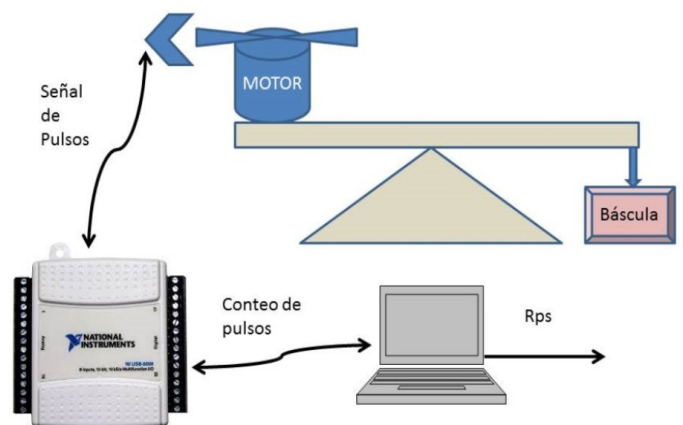


Figura 4: Banco experimental propuesto para caracterización de fuerza de los rotores.

Tabla 1: Pérdida de empuje de los rotores de acuerdo a una pérdida parcial de la hélice.

Reducción en la hélice	Empuje	Pérdida
0 cm	38.53 N	0 %
1 cm	30.04 N	22 %
2 cm	22.46 N	41.7 %

4.2. Detección de fallas basada en datos de la IMU

Los experimentos para la detección de fallas se realizaron, por seguridad, sobre una plataforma mecánica de tipo giroscopio de la empresa Eureka Dynamics, referida como FFT-GYRO (Valencia-Palomo et al., 2018) que se encuentra instrumentada con un sistema de cardanes que permiten el desplazamiento rotacional del vehículo en los tres ejes y evitando su desplazamiento traslacional. El FFT-GYRO se muestra en la Figura 5.

Una vez montado el vehículo sobre la plataforma, se obtuvieron m cantidad de datos correspondientes a la orientación en su representación en cuaterniones (w_q, x_q, y_q, z_q) , la velocidad angular de rotación sobre cada uno de sus ejes (x_v, y_v, z_v) así como su aceleración lineal (x_a, y_a, z_a) operando en condiciones nominales. Para realizar esta tarea fue necesario obtener los datos proporcionados por la unidad de medición inercial (IMU) del VANT conectándolo de forma inalámbrica a un ordenador. A través del ordenador, mediante ROS, es posible conectarse al ROSMASTER de ErleBrain 3, y a través de la publicación de sus tópicos, obtener los datos de la IMU a través de `ROSTOPIC/MAVROS/IMU/DATA`. La publicación de este tópico permite obtener los datos que se requirieron.

Con estos datos se construye la matriz de mediciones $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_{10}] \in \mathbb{R}^{m \times 10}$, donde cada vector representa la información de los datos anteriormente especificados. En la etapa de entrenamiento la matriz de datos \mathbf{X} , se procesa mediante PCA para extraer las características principales del sistema y la reducción de dimensionalidad de los datos. Una vez obtenidos las componentes principales, y caracterizada la región de operación nominal, se toman a éstas como la *nueva base* de los datos extraídos. De igual manera se sigue con el procedimiento mostrado en la sección anterior donde se induce falla en cada uno de los rotores a través del recorte de 2 cm a la hélice. Sin romper la hélice se coloca nuevamente en su posición y se simula el vuelo del VANT en modo estacionario montado sobre la plataforma FFT-GYRO. Debido al recorte y la velocidad angular generada por los rotores, la hélice se desprende creando el efecto de falla. Con las señales obtenidas se crea una base de datos de las fallas en cada una de las hélices. Se repiten los casos de estudio de la Figura 6. A cada caso se le aplica el PCA para la extracción de características en términos de la nueva base, considerando la media y la desviación estándar de los datos nominales. Con ello se crea una matriz de datos que contiene los datos nominales, y los datos con fallas, que fueron procesados con PCA.

Los datos de la matriz que contiene la extracción de características serán ahora las nuevas entradas para el clasificador en la etapa de entrenamiento, que se aplicará fuera de línea, dado que ambos pertenecen a algoritmos de aprendizaje de máquina supervisados, deben ser previamente etiquetados. Para la etapa de entrenamiento se trabajó con el 80 % de los datos obtenidos y el restante 20 % se utilizó para la etapa de diagnóstico.



Figura 5: Plataforma mecánica FFT-GYRO.

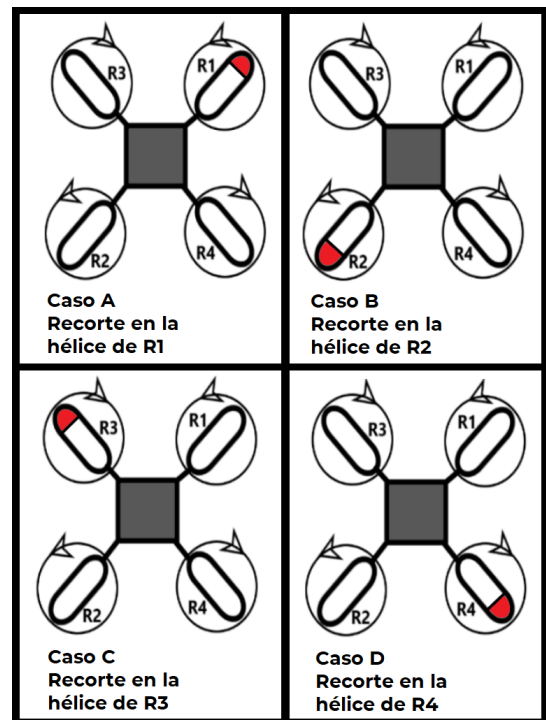


Figura 6: Casos de estudio de fallas en los rotores del vehículo por el recorte de las hélices.

Con el objetivo de ejemplificar cómo se aprecian los datos, en la Figura 7 se muestran los datos de vuelo proporcionados por la IMU cuando se desprende la hélice del rotor r_4 en $t = 223$ s. Nótese que a simple vista no es posible apreciar un gran efecto del desprendimiento de la hélice, esto es debido a

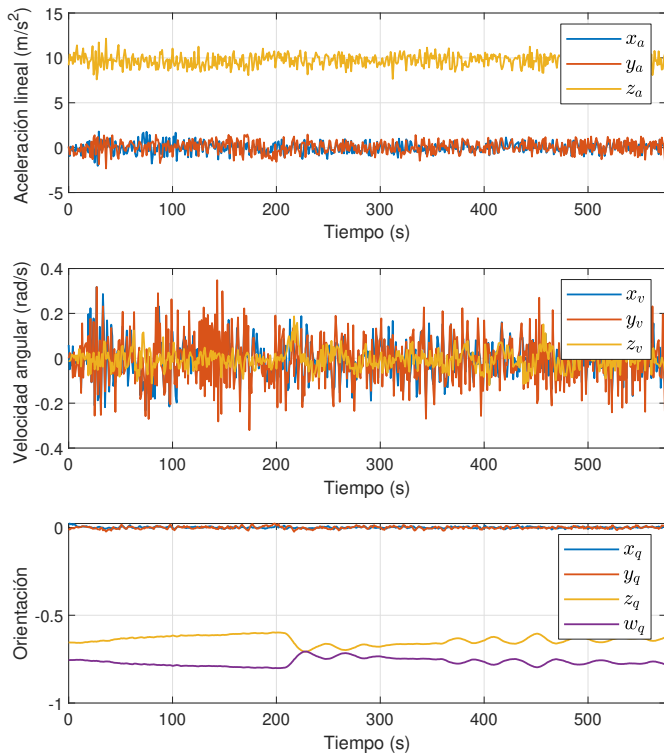


Figura 7: Datos del VANT en vuelo estacionario cuando presenta falla en la hélice del rotor 4.

que el control de vuelo compensa la fuerza de empuje pérdida con los tres rotores restantes. En un escenario de vuelo real este efecto se podría apreciar con una variación súbita en los ángulos de rotación acompañado de la deriva en la traslación y pérdida de altura. Seguido por la recuperación de su posición rotacional y traslacional por la acción del controlador. Sin embargo, si la falla no se detecta podría llegar a desestabilizar el vehículo llevando a generar accidentes que no solo dañarían el VANT, sino también podría ocasionar daños en el entorno o incluso a personas. Es aquí donde se aprecia la importancia de contar con un algoritmo de tratamiento de datos como el PCA que busca describir un conjunto de datos en términos de nuevas variables no correlacionadas.

5. Resultados

Una vez obtenido los datos del VANT cuadricóptero en condiciones normales y en condiciones de fallas en los rotores r_1 , r_2 , r_3 y r_4 se extraen los datos y se normalizan. Es decir, se les aplica una distribución normal con media cero y desviación estándar en 1, para posteriormente obtener la matriz de puntuaciones estandarizadas y se grafican en términos de la nueva base correspondiente a las componentes principales tal como se explica en la Sección 2. La varianza total del conjunto de datos es $\lambda_1 + \lambda_2 + \dots + \lambda_{10} = 2.44 + 1.94 + 1.23 + 1.09 + 1.04 + 0.91 + 0.77 + 0.36 + 0.18 + 0.001 = 10$. Estos datos corresponden a la varianza explicada por cada componente y se muestra de manera gráfica en el diagrama de Pareto de la Figura 8. Si se quisiera reducir la dimensionalidad de los datos, como ejemplo, podrían despreciarse las últimas 3 componentes principales y con ello se tendría el 94.47 % de la variabilidad de los datos.

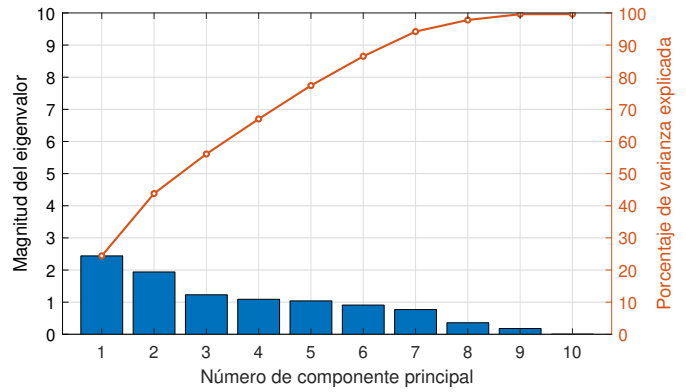


Figura 8: Diagrama de Pareto que muestra la contribución de cada componente principal en la variabilidad de los datos.

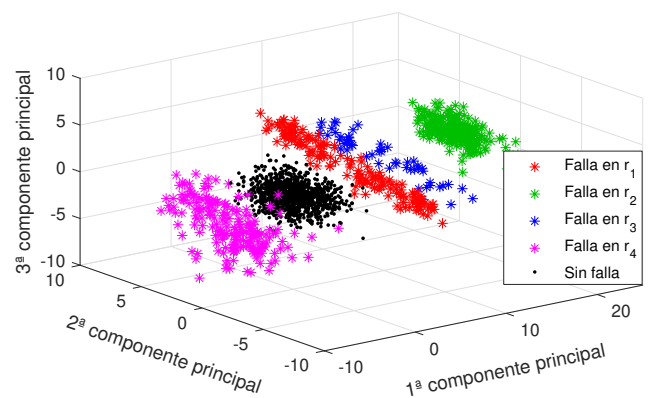


Figura 9: Mediciones proyectadas en el subespacio PCA tridimensional.

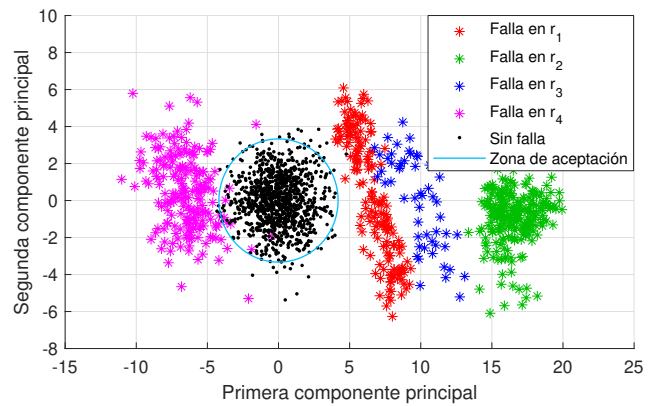


Figura 10: Mediciones proyectadas en el subespacio PCA bidimensional.

Con el modelo PCA definido por $\{P, \Lambda, \mu, \sigma\}$ es posible inferir si cualquier medición $\mathbf{x}_{new} = [x_1 \ x_2 \ \dots \ x_n]^T$ es compatible con el *comportamiento normal* del proceso. Para ello se estandariza \mathbf{x}_{new} con las medias y desviaciones estándar del modelo nominal. Luego la muestra estandarizada \mathbf{z}_{new} puede ser expresada en términos de la base ortonormal \mathbf{P} con las coordenadas de (14). Para las mediciones obtenidas (\mathbf{x}_{new}) con las fallas en los actuadores de los rotores r_1 , r_2 , r_3 y r_4 , se normalizan (\mathbf{z}_{new}) y se proyectan en el subespacio vectorial de las componentes principales. Esta proyección se observa en la Figura 9 en términos de las 3 primeras componentes principales y en la Figura 10

una proyección ortonormal de la misma en términos de la primera componente principal (P_1) y segunda componente principal (P_2), cuyo objetivo es verificar la separabilidad de los datos con fallas y sin fallas (dentro de la zona de operación nominal).

Una manera de determinar la existencia de fallas es comparar la nueva muestra transformada (\mathbf{t}_{new}) con los datos proyectados correspondientes a la región nominal. Así los datos que caen fuera de la región nominal, representan parámetros anormales como se muestra en la Figura 10.

En las Figuras 9 y 10 se aprecian zonas de operación de fallas con forma diferente, esto ocurre porque en la práctica el VANT no es totalmente simétrico y el cambio de la representación de los datos puede hacerlo más evidente. En la Figura 10 se aprecia que las mediciones correspondientes a fallas cuando se mapean al subespacio principal aparecen separadas del grupo correspondiente a la proyección de las mediciones sin falla. De esta manera, el índice T^2 resulta útil para detectar la presencia de fallas. En la Figura 11 se muestra una gráfica del monitoreo del estadístico T^2 durante 575 s de *vuelo* en la plataforma FFT-GYRO. El umbral $U_{T^2} = 7.82$, correspondiente al nivel de confianza $\alpha = 0.98$, se calculó mediante (19) para dos componentes principales. La gráfica de T^2 en la Figura 11 fue obtenido con las componentes principales de los datos en la Figura 7 que corresponden a una falla del rotor r_4 en el instante $t = 223$ s. Una particularidad en este caso es que, aunque la falla (rotura parcial de la hélice) permanece para $t \geq 223$ s, sólo es detectable durante un periodo corto, debido a que el controlador de vuelo y la plataforma giroscópica restablecen la orientación del VANT. Por otra parte, la detección de la falla mediante esta técnica no aporta información sobre el origen o ubicación de la falla (no se sabe cuál rotor falló), de ahí la propuesta de utilizar métodos que *aprendan* a identificar la ubicación de la falla.

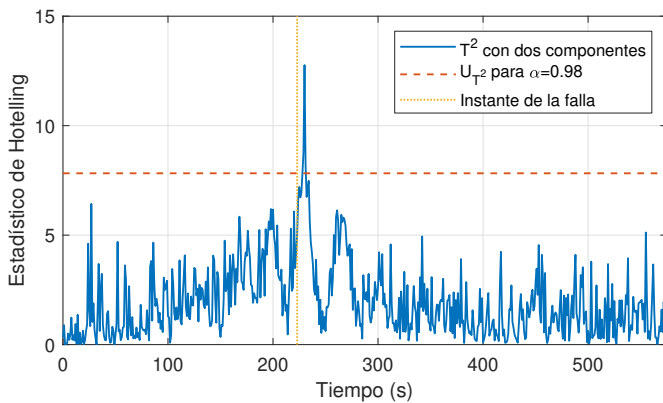


Figura 11: Detección de falla en uno de los rotores mediante el monitoreo del estadístico T^2 .

La separación de las fallas en múltiples clases se dificulta partiendo sólo de T^2 , por ello en lugar de combinar las componentes principales en un solo estadístico éstas se usan como variables de entrada (*features*) de un algoritmo de aprendizaje automático que permita detectar las fallas y separarlas por su origen o localización.

Como entradas de los clasificadores se tomaron las primeras siete componentes principales de las diez variables reportadas por la IMU del VANT. En este trabajo se utilizaron dos clasificadores, k -NN y SVM, para evaluar cuál ofrecía un mejor des-

empeño. Los datos de entrada para ambos algoritmos se organizaron en vectores $\mathbf{x} = [PC_1, PC_2, \dots, PC_7]$, donde las PC_i son las componentes principales. Cada dato del conjunto de entrenamiento se etiquetó como perteneciente a una de cinco clases, c_1, c_2, \dots, c_5 . Como etiquetas se usaron enteros positivos indicando la clase a la que pertenece cada dato. La clase c_1 corresponde a los datos de operación sin falla, y las clases c_2, \dots, c_5 corresponden a los datos de operación con falla en los rotores r_1, \dots, r_4 . Para la fase de entrenamiento se eligió un 80 % de los datos y para la fase de prueba un 20 %, como se observa en la Tabla 2.

Tabla 2: Número de datos para las fases de entrenamiento y prueba.

Clase	Descripción	Total	Entrenamiento	Prueba
1	Sin falla	1085	868	217
2	Falla en r_1	276	221	55
3	Falla en r_2	231	185	46
4	Falla en r_3	151	121	30
5	Falla en r_4	376	301	75

Para evaluar los algoritmos, con el software MATLAB® se calcularon los índices de pérdidas por resustitución (*resubLoss*, del inglés *resubstitution loss*), como una forma de medir la incapacidad de los clasificadores para clasificar correctamente las fallas conocidas dentro del conjunto de entrenamiento. Para el algoritmo k -NN se encontró que se obtenía una mejor respuesta con un número $k = 3$ de vecinos más cercanos, alcanzando un índice *resubLoss* = 0.0012. El algoritmo SVM obtuvo un mejor desempeño para la clasificación de los datos de entrenamiento, logrando clasificar correctamente todas las muestras, por lo que se obtuvo un índice *resubLoss* = 0.

También se evaluó el desempeño de ambos algoritmos con los datos de prueba. La métrica que se utilizó para cuantificar el desempeño en la detección de fallas es la exactitud del clasificador (ACC, del inglés *accuracy*), debido a que esta métrica considera no sólo los aciertos al detectar las fallas existentes sino que también penaliza los errores debido a la detección de fallas inexistentes. Para el algoritmo k -NN, el desempeño se resume en la matriz de confusión de la Tabla 3, donde se clasificaron el 89.6 % de los datos correctamente (ACC = 0.896). Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Es decir, la diagonal principal representa el número de instancias correctamente clasificadas en cada clase, y fuera de ella, las que no se clasificaron correctamente. Para el algoritmo SVM el desempeño se observa en la Tabla 4, donde se clasificaron el 92.4 % de los datos correctamente (ACC = 0.924). Si bien, en general, con el SVM se obtuvo una mayor exactitud de clasificación (menor índice de pérdidas), el algoritmo k -NN produjo una mejor separación de las clases de falla, aunque tiende a producir algunos falsos positivos (detecta fallas que no existen). En cambio, el SVM no produjo falsos positivos, pero tuvo cierta dificultad para distinguir entre los diferentes tipos de falla. Respecto a la exhaustividad (REC, del inglés *recall*) de la detección de fallas, el SVM también mostró un desempeño superior, alcanzando un valor de REC = 0.98, mientras que el k -NN sólo alcanzó el valor de REC = 0.82.

Tabla 3: Matriz de confusión del algoritmo k -NN con los datos de prueba.

1	183				44	80.6%	19.4%
2		55				100.0%	
3			46			100.0%	
4		1		29		96.7%	3.3%
5					75	100.0%	
	100.0%	98.2%	100.0%	100.0%	63.0%		
		1.8%					37.0%
	1	2	3	4	5		
	Clase predicha						

Tabla 4: Matriz de confusión del algoritmo SVM con los datos de prueba.

1	217					100.0%	
2		55				100.0%	
3	11		35			76.1%	23.9%
4	1			26	3	86.7%	13.3%
5	17				58	77.3%	22.7%
	88.2%	100.0%	100.0%	100.0%	95.1%		
	11.8%						4.9%
	1	2	3	4	5		
	Clase predicha						

6. Conclusiones

En este trabajo se abordó el problema de la detección y clasificación de fallas en un VANT del tipo cuadirrotor bajo un enfoque basado en el análisis de datos. El argumento central de este enfoque fue el de seleccionar los datos más adecuados para que, a través del análisis en componentes principales se realizaran dos tareas fundamentales: extraer las características más importantes de los datos y reducir las dimensiones de los vectores de datos. Con esto fue posible obtener *a posteriori* una clasificación adecuada. En la prueba experimental se analizaron los datos obtenidos de la unidad inercial del VANT. Al realizar el análisis PCA se observa una separación entre las regiones de operación nominal y las que presentan fallas en las hélices de los rotores. Para la etapa de diagnóstico se consideraron dos técnicas de clasificación: k -NN y SVM. El algoritmo SVM, para este sistema en específico, obtuvo un mejor desempeño que el k -NN con los datos de prueba. Esto se debe a que las componentes principales de la clase sin falla presentan una buena separación respecto a las clases con falla, pero la separación entre las diferentes clases de falla es menos clara. La menor separabilidad de las clases de falla con el SVM puede atribuirse a la naturaleza lineal de este método, por lo que a futuro se explorará su desempeño con kernels no lineales. En general, los resultados demuestran que la combinación de la técnica estadística PCA con el clasificador SVM es el más adecuado para afrontar el problema de la detección y diagnóstico de fallas en actuadores de un VANT.

En el trabajo a futuro se explorarán otras metodologías del aprendizaje automático aplicando por ejemplo redes neuronales artificiales para la identificación de otro tipo de fallas como bloqueos, daños en la estructura del vehículo, fallas en sensores, entre otros. Asimismo, también es posible que se pueda complementar al algoritmo con nuevas variables que pudieran ser importantes, por ejemplo, la corriente demandada por los rotores o la temperatura en cada rotor. La metodología propuesta puede aplicarse a diferentes tipos de vehículos aéreos a través de plataformas experimentales o bases de datos como las propuestas en Keipour et al. (2019). Finalmente, es importante destacar que el método propuesto puede considerarse como una herramienta perfecta para poder obtener un modelo experimental mucho más preciso, del sistema a emplear. La mejora en la exactitud del modelo amplía las posibilidades de minimizar o reducir los esfuerzos en las estrategias de control tolerante a fallas a diseñar en futuros trabajos, donde se tiene el interés particular en la utilización de algoritmos de control predictivo eficiente (e.g. Khan et al., 2011) que puedan ser embebidos en plataformas de vuelo.

Referencias

- Alos, A., Dahrouj, Z., 2020. Detecting contextual faults in unmanned aerial vehicles using dynamic linear regression and k -nearest neighbour classifier. *Gyroscopy and Navigation* 11, 94–104.
- Baskaya, E., Bronz, M., Delahaye, D., 2017. Fault detection & diagnosis for small uavs via machine learning. in: *Digital Avionics Systems Conference (DASC)*, 2017 IEEE/AIAA 36th, IEEE. pp. 1–6.
- Benini, A., Ferracuti, F., Monteriù, A., Radensleben, S., 2019. Fault detection of a VTOL UAV using acceleration measurements. in: *2019 18th European Control Conference (ECC)*, IEEE. pp. 3990–3995.
- Freeman, P., Pandita, R., Srivastava, N., Balas, G.J., 2013. Model-based and data-driven fault detection performance for a small UAV. *IEEE/ASME Transactions on Mechatronics* 18, 1300–1309.
- Gertler, J., 2015. Fault detection and diagnosis. *Encyclopedia of Systems and Control*, 417–422.
- Ghalamchi, B., Mueller, M., 2018. Vibration-based propeller fault diagnosis for multicopters. in: *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE. pp. 1041–1047.
- Guo, K., Liu, L., Shi, S., Liu, D., Peng, X., 2019. UAV sensor fault detection using a classifier without negative samples: A local density regulated optimization algorithm. *Sensors* 19, 771.
- Guzmán-Rabasa, J.A., López-Estrada, F.R., González-Contreras, B.M., Valencia-Palomo, G., Chadli, M., Perez-Patricio, M., 2019. Actuator fault detection and isolation on a quadrotor unmanned aerial vehicle modeled as a linear parameter-varying system. *Measurement and Control* 52, 1228–1239.
- Iannace, G., Ciaburro, G., Trematerra, A., 2019. Fault diagnosis for UAV blades using artificial neural network. *Robotics* 8, 59.
- Jiang, Y., Zhiyao, Z., Haoxiang, L., Quan, Q., 2015. Fault detection and identification for quadrotor based on airframe vibration signals: a data-driven method. in: *2015 34th Chinese Control Conference (CCC)*, IEEE. pp. 6356–6361.
- Jolliffe, I., 2011. *Principal component analysis*. Springer.
- Keipour, A., Mousaei, M., Scherer, S., 2019. Alfa: A dataset for UAV fault and anomaly detection. *arXiv preprint arXiv:1907.06268*.
- Khan, B., Rossiter, J.A., Valencia-Palomo, G., 2011. Exploiting kautz functions to improve feasibility in MPC. *IFAC Proceedings Volumes* 44, 6777–6782.
- Li, M., Li, G., Zhong, M., 2016. A data driven fault detection and isolation scheme for UAV flight control system. in: *Control Conference (CCC)*, 2016 35th Chinese, IEEE. pp. 6778–6783.
- López-Estrada, F.R., Rotondo, D., Valencia-Palomo, G., 2019. A review of convex approaches for control, observation and safety of linear parameter varying and Takagi-Sugeno systems. *Processes* 7, 814.
- López-Estrada, F.R., Santos-Estudillo, O., Valencia-Palomo, G., Gómez-Peñate, S., Hernandez-Gutiérrez, C., 2020. Robust qLPV tracking fault-tolerant control of a 3 dof mechanical crane. *Mathematical and Computational Applications* 25, 48.

- Martinez, W.L., Martinez, A.R., 2015. Computational statistics handbook with MATLAB. Chapman and Hall/CRC.
- Mouloua, M., Gilson, R., Kring, J., Hancock, P., 2001. Workload, situation awareness, and teaming issues for UAV/UCAV operations, in: Proceedings of the human factors and ergonomics society annual meeting, SAGE Publications Sage CA: Los Angeles, CA. pp. 162–165.
- Mueller, M.W., D'Andrea, R., 2014. Stability and control of a quadcopter despite the complete loss of one, two, or three propellers, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 45–52.
- Nonami, K., Kendoul, F., Suzuki, S., Wang, W., Nakazawa, D., 2010. Introduction, in: Autonomous Flying Robots. Springer, pp. 1–29.
- Qin, S.J., 2012. Survey on data-driven industrial process monitoring and diagnosis. Annual reviews in control 36, 220–234.
- Russell, E.L., Chiang, L.H., Braatz, R.D., 2012. Data-driven methods for fault detection and diagnosis in chemical processes. Springer Science & Business Media.
- Saied, M., Lussier, B., Fantoni, I., Francis, C., Shraim, H., Sanahuja, G., 2015. Fault diagnosis and fault-tolerant control strategy for rotor failure in an octo-rotor, in: IEEE International Conference on Robotics and Automation, IEEE. pp. 5266–5271.
- Santos-Ruiz, I., López-Estrada, F.R., Puig, V., Blesa, J., Javadiha, M., 2019. Localización de fugas en redes de distribución de agua mediante k -NN con distancia cosenoidal. Asociación de México de Control Automático .
- Sharifi, F., Mirzaei, M., Gordon, B.W., Zhang, Y., 2010. Fault tolerant control of a quadrotor uav using sliding mode control, in: 2010 Conference on Control and Fault-Tolerant Systems (SysTol), IEEE. pp. 239–244.
- Strang, G., Strang, G., Strang, G., Strang, G., 2016. Introduction to linear algebra. volume 3. Wellesley-Cambridge Press Wellesley, MA.
- Sun, R., Cheng, Q., Wang, G., Ochieng, W., 2017. A novel online data-driven algorithm for detecting UAV navigation sensor faults. Sensors 17, 2243.
- Tamura, M., Tsujita, S., 2007. A study on the number of principal components and sensitivity of fault detection using PCA. Computers & Chemical Engineering 31, 1035–1046.
- Valencia-Palomo, G., Villanueva-Grijalba, O., Robles-Ríos, R., 2018. Device for the pose measurement and test of control algorithms for unmanned aerial vehicles. Mexican Patent MX/a/2017/005377.
- Vapnik, V., 2013. The nature of statistical learning theory. Springer Science & Business Media.
- Vey, D., Lunze, J., 2016. Experimental evaluation of an active fault-tolerant control scheme for multirotor uavs, in: 2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol), IEEE. pp. 125–132.
- Wang, B., Peng, X., Jiang, M., Liu, D., 2020. Real time fault detection for UAV based on model acceleration engine. IEEE Transactions on Instrumentation and Measurement .
- Wang, B., Wang, Z., Liu, L., Liu, D., Peng, X., 2019. Data-driven anomaly detection for UAV sensor data based on deep learning prediction model, in: 2019 Prognostics and System Health Management Conference (PHM-Paris), IEEE. pp. 286–290.
- Wold, S., 1978. Cross-validators estimation of the number of components in factor and principal components models. Technometrics 20, 397–405.
- Xian, B., Hao, W., 2019. Nonlinear robust fault-tolerant control of the tilt tri-rotor UAV under rear servo's stuck fault: Theory and experiments. IEEE Transactions on Industrial Informatics 15, 2158–2166.
- Xiao, K., Zhao, J., He, Y., Li, C., Cheng, W., 2019. Abnormal behavior detection scheme of UAV using recurrent neural networks. IEEE Access 7, 110293–110305.
- Yang, H., Meng, C., Wang, C., 2020. A hybrid data-driven fault detection strategy with application to navigation sensors. Measurement and Control , 0020294020920891.
- Yap, Y.K., 2014. Structural health monitoring for unmanned aerial systems. EECS., UNC, BerNley, Rep. UCB/EECS-2014-70 .
- Yousefi, P., Fekriazgomi, H., Demir, M.A., Prevost, J.J., Jamshidi, M., 2018. Data-driven fault detection of un-manned aerial vehicles using supervised learning over cloud networks, in: 2018 World Automation Congress (WAC), IEEE. pp. 1–6.