

## Estimación de orientación de un vehículo aéreo no modelado usando fusión de sensores inerciales y aprendizaje de máquina

Ruben Fonnegra<sup>a</sup>, German Goetz<sup>b</sup>, Andrés Tobón<sup>b,\*</sup>

<sup>a</sup>Institución Universitaria Pascual Bravo, Medellín, Colombia, 050036

<sup>b</sup>Instituto Tecnológico Metropolitano (ITM), Medellín, Colombia, 050012

### Resumen

Los vehículos aéreos no tripulados (UAV) ofrecen alternativas para diversas aplicaciones en las que se compromete la integridad humana. En este sentido, la necesidad de incrementar la autonomía de estos vehículos presenta una alternativa al área de inteligencia artificial para aumentar las capacidades de navegación en diversos entornos. Este artículo presenta una evaluación para estimación de inclinación y orientación, utilizando algoritmos de aprendizaje automático para una planta dinámica con múltiples rotores. Para esto se propone un experimento para recopilar datos de unidades de medición inercial (IMU) sobre la placa de un UAV, y sometidos a diferentes inclinaciones antes de lograr la tarea de clasificación. Los resultados reportados usando los algoritmos de  $k$  vecinos más cercanos ( $k$ -NN), máquinas de soporte vectorial (SVM) y de Bayes muestran eficiencia en el reconocimiento, obteniendo una precisión hasta del 99 %. Además, estos algoritmos podrían combinarse con técnicas de control robustas, ideal para la implementación en sistemas con capacidades de procesamiento limitadas.

### Palabras Clave:

Sensores inerciales, Inteligencia artificial, aprendizaje de máquinas., UAV

### Orientation estimating in a non-modeled aerial vehicle using inertial sensor fusion and machine learning techniques

### Abstract

Unmanned Aerial Vehicles (UAV) have offered alternatives for applications in which human integrity is compromised. In this sense, the need of increasing autonomy in these vehicles presents an alternative to artificial intelligence areas to enhance navigation capacities through several environments. This article presents an evaluation for estimating inclination and orientation, using automatic learning algorithms for a dynamic multi-rotor plant. To do so, an experiment is proposed to collect the data from multiple IMU sensors over an UAV main board, and submitted to different inclinations before achieving the classification task. The reported results using  $k$  nearest neighbors ( $k$ -NN), support vector machines (SVM) and Bayes show efficiency during the recognition, obtaining an accuracy score up to 99 %. Besides, the algorithms could be combined along with robust control techniques, which is ideal for implementation in embedded systems with low processing capacities.

### Keywords:

Inertial Sensors, Artificial Intelligence, Machine Learning, UAV,

## 1. Introducción

Desde que en la década de los 40 del siglo XX se usaron los primeros dispositivos que de alguna forma navegaban de forma autónoma. A partir de allí se han desarrollado múltiples plataformas con diferentes usos en los cuales predomina la necesi-

dad de controlar el vehículo de forma remota. En este contexto observamos el envío de naves espaciales controladas desde la tierra, el uso de vehículos aéreos en misiones de vigilancia y en algunos casos de combate. Recientemente, se han propuesto nuevas aplicaciones y usos de los vehículos aéreos no tripulados

\*Autor para correspondencia: andrestobon@itm.edu.co

(UAVs por sus siglas en Inglés), entre las cuales se encuentran los sistemas de vigilancia, monitoreo de campos de cultivos, operativos de búsqueda o rescate entre otros. Estas aplicaciones demandan robustez en los sistemas computacionales y de control en los UAVs con el fin de incrementar la seguridad y confiabilidad en la implementación de los sistemas de navegación de este tipo de dispositivos, el cual está directamente relacionado con la capacidad del vehículo de percibir el medio sobre el cual se desplaza.

Este trabajo es la continuación de una serie de trabajos realizados por los autores (Góez, 2016) (Darío et al., 2014) (Colmenares-Quintero and Góez-Sánchez, 2018), los cuales abordan la navegación de vehículos aéreos no tripulados basados en sistemas multi-rotor, en busca de técnicas para fortalecer los sistemas de navegación, con el fin de disminuir el riesgo de impacto o colisión al momento de realizar vuelos en diversos entornos en los cuales se obstaculiza la ruta de vuelo. En este sentido se encuentran trabajos que abordan el problema de navegación desde la robustez de los controladores ubicados en sistemas no embebidos, los cuales utilizan sistemas de comunicación inalámbrica para recibir información desde los sensores, con el fin de estimar o corregir parámetros esenciales para la estimación de trayectorias o la corrección en la información que se le envía al controlador embebido implementado sobre el UAV (Tarazona et al., 2015). Los sistemas de navegación embebidos usan unidades de medición inercial (por sus siglas en inglés, IMU), con el fin de medir las fuerzas básicas (tales como aceleración, velocidad angular, campo magnético, entre otros) para estimar adecuadamente la inclinación, orientación y velocidad de giro del vehículo (Li et al., 2014; Wendel et al., 2006). La realimentación de los datos obtenidos del sistema de medición inercial, más el modelo dinámico del UAV constituyen en el insumo básico para diseñar un sistema de control clásico o moderno, el cual cumplirá la función de manipular automáticamente el vehículo (Zul Azfar and Hazry, 2011). En este sentido, deben considerarse diversos factores fundamentales para la eficiencia y desempeño del sistema, tales como el tiempo de procesamiento (dada la limitada capacidad de las plataformas embebidas) además de las capacidades computacionales de los controladores.

Un aspecto importante previo al diseño del controlador es el tratamiento de los datos provenientes de las unidades de medición, debido a que se deben identificar las variables externas que intervienen directamente en el comportamiento del sistema, tales como la velocidad del viento y su resistencia al avance, el peso específico de la aeronave o su capacidad de sustentación. Bajo este esquema se identificaron trabajos que proponen métodos para estimar y corregir la afectación que causan este tipo de perturbaciones en la navegabilidad de un UAV usando una IMU. En (Song et al., 2016), los autores presentan un método de corrección de la inclinación en función de la cinemática del sistema, la influencia del viento y considerando la velocidad y el ángulo de inclinación de la planta. Sin embargo, el modelo propuesto tiene en cuenta los parámetros dinámicos del vehículo, además de la utilización de un controlador en espacio de estados para la corrección de la inclinación basada en un modelo de estimación del viento. Esta consideración permite que el modelo en simulación obtenga resultados adecuados, sin embargo como mencionan los autores, la corrección podría ser imprecisa

en su implementación sobre una plataforma real.

Los métodos control moderno se basan en el análisis y síntesis en el dominio del tiempo mediante variables de estado. De esta forma, los filtros de Kalman ha sido una estrategia ampliamente explorada para determinar la orientación de diferentes sistemas, mediante la fusión de diferentes fuentes de información. Esto, con el objetivo de determinar las variables de estado de los sistemas sometidos a ruido blanco característico en la utilización de sistemas inerciales, además de estimar el comportamiento de la planta y generar un horizonte de predicción que contribuya en el diseño de un controlador estable (Von Chong and Cabañero, 2014).

En (Goslinski et al., 2013), se describe la aplicación de un filtro Kalman en un módulo de orientación en un vehículo aéreo no tripulado, comparado con un encoder de referencia acoplado a un sistema embebido de baja gama, que busca estimar orientación en tiempo real. A pesar de que la estrategia propuesta por los autores obtiene buen desempeño en sistemas con perturbaciones leves para la corrección de la inclinación, pierde precisión al momento de introducir perturbaciones significativas. Esto se podría corregir mediante la estimación de los estados dinámicos del sistema, o utilizando un algoritmo computacional para aprender el comportamiento de la planta durante perturbaciones. En (Lupashin, 2011) ofrecen un método utilizado para que un vehículo aéreo no tripulado golpee una pelota hacia un objetivo, para esto su algoritmo de generación de trayectorias utiliza un filtro Kalman, similar a (Goslinski et al., 2013; Razak et al., 2013; Goez et al., 2016), donde se implementa un controlador de posición y altitud además de un filtro Kalman para el control de la orientación.

A pesar de encontrar trabajos que proponen técnicas para corregir la inclinación del UAV en función de las perturbaciones del medio, se encontró que los métodos usados para corregir la inclinación requieren conocer el modelo dinámico de la plataforma. Esto los convierte en sistemas diseñados para casos específicos, y en el caso de sufrir variaciones en el diseño físico, se hace necesaria la implementación de un nuevo diseño de controlador a partir de un nuevo modelo cinemático y los parámetros dinámicos de los mismos.

En este sentido, diversos trabajos en el estado de arte proponen la utilización de estrategias basadas en aprendizaje computacional con el fin de mejorar los sistemas de navegación ante este tipo de inconvenientes (Kan et al., 2013; Giusti et al., 2016). Proponer nuevas técnicas que se comporten como sistemas adaptativos y que se adecuen fácilmente a los cambios físicos de la planta ayudará a disminuir el tiempo de desarrollo de este tipo de sistemas, además de disminuir los riesgos asociados a la falta de experticia de algunos constructores aficionados de UAVs, que por no contar con los suficientes conocimientos en diseño de controladores pueden poner en riesgo el equipo y al entorno de trabajo, tales como personas, animales o bienes inmuebles. En este trabajo se propone una estrategia para estimar la inclinación de un UAV del que se desconozca su modelo cinemático, con el fin de adaptar el sistema dinámico de orientación a algoritmos de control y guiados, basados en técnicas de aprendizaje computacional (Goez et al., 2016). La estrategia propuesta consiste en implementar técnicas de clasificación de datos tomados de la unidad de medición inercial, en donde la unidad se implementa directamente sobre el UAV multi-rotor, permitien-

do obtener una base de datos que representa las características dinámicas del UAV en términos de aceleración gravitatoria y velocidad de rotación. Este trabajo se presenta de la siguiente manera: En la sección 2 se describen los elementos utilizados para la extracción de la base de datos, el preprocesamiento, filtrado y la implementación de los algoritmos computacionales. En la sección 3 se presenta la descripción de los experimentos realizados, incluyendo los resultados obtenidos. Por último, en la sección 4 se presentan las principales conclusiones del trabajo, además de los trabajos futuros a realizar.

## 2. Materiales y Métodos

En la Figura 1 se muestra de forma general, las etapas empleadas para la clasificación de las inclinaciones obtenidas a partir de los sensores inerciales. En primera instancia, se propone una etapa de adquisición de datos, en la cual se obtiene una base de datos experimental que contiene datos correspondientes a la aceleración angular y velocidad angular obtenidos de una IMU. Para esto, se propone un experimento para adquirir datos sobre la plataforma principal de un UAV. A continuación se implementa una etapa de limpieza de las señales usando el filtro de Kalman, con el fin de eliminar el ruido blanco gaussiano, característico de este tipo de señales. Después, se implementa una etapa de aprendizaje usando diferentes técnicas de aprendizaje a partir de los datos filtrados provenientes del sensor. Los modelos utilizados comprenden los más utilizados en la literatura, en términos de eficiencia computacional y eficacia en cuanto a clasificación (*Knn*, *SVM*, *Bayes*). Además, de cara a la implementación sobre una plataforma computacional limitada, también se considera el tiempo computacional empleado para clasificar las instancias entrantes (datos de los sensores). Una vez entrenados los modelos, se extraen métricas ampliamente utilizadas en el estado del arte (precisión y matrices de confusión) a partir de validación cruzada, dada su relevancia estadística teniendo en cuenta la variación de los datos en la entrada. Para concluir se presentan los resultados con las medidas obtenidas durante todas las iteraciones de la validación cruzada.

### 2.1. Etapa de adquisición, Base de datos y preprocesamiento

Para la implementación de los algoritmos se obtuvo una base de datos experimental que contiene la información tomada de un acelerómetro y un giróscopo (IMU) instalados sobre la plataforma de un UAV. A esta plataforma se le modifica la posición y orientación con el fin de determinar en ella una inclinación para realizar la medición. A partir de allí, se adquirieron los datos correspondientes a la aceleración y velocidad del giro en los tres ejes para la plataforma en una posición completamente horizontal (6 grados de libertad), con inclinación a la derecha, a la izquierda, hacia adelante y hacia atrás. También se consideraron posiciones intermedias entre las mencionadas anteriormente, tales como derecha - adelante, izquierda - adelante, derecha - atrás e izquierda - atrás.

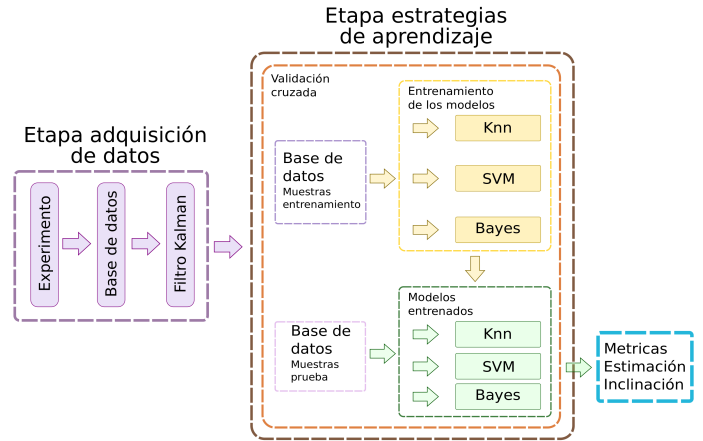


Figura 1: Metodología propuesta Puls and Hein (2010)

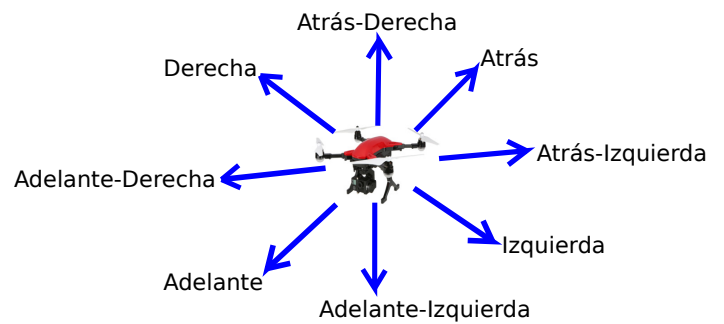


Figura 2: Caracterización de las posiciones del UAV para la extracción de los datos (fuente: Autor).

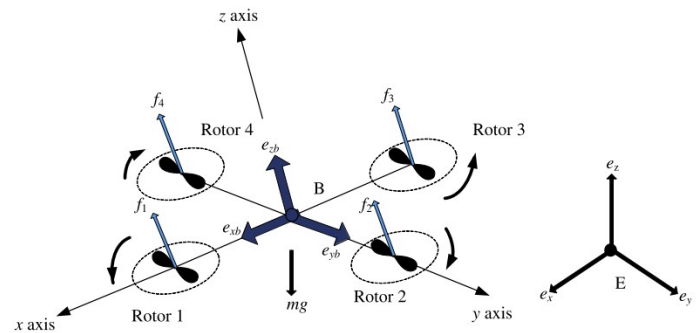


Figura 3: Diagrama de par y fuerzas asociados a un quadcopter Puls and Hein (2010)

En la Figura 2, el esquema de direcciones consideradas para el sistema, está relacionado con las posibles direcciones que puede tomar el UAV en función de la orientación permitida por la geometría del mismo y la configuración de los rotores. Por lo tanto, la inclinación produce un cambio en las fuerzas rotacionales y de desplazamiento. En la Figura 3, se observa cómo la velocidad y dirección de los rotores afectan la inclinación y las fuerzas asociadas al desplazamiento. Siendo evidente que existe una relación de magnitud física entre la inclinación del vehículo y la velocidad del mismo, por lo que es posible estimar la velocidad a partir del ángulo de inclinación de los rotores.

Con el fin de construir una base de datos que contenga información semejante a la obtenida durante vuelo en tiempo real utilizando elementos de bajo costo, se considera la utilización del sensor GY-521. Este dispositivo posee un chip MPU6050, el cual contiene un acelerómetro y giróscopo micro electro-mecánico (MEMS (Pieters, 2009)); ambos de tres ejes correspondientes a los planos cartesianos (X, Y, Z). Sin embargo, la utilización de este dispositivo en comparación con otros de igual clasificación yace en la pérdida de precisión derivada de su proceso de fabricación dado que es de bajo costo. Es por esto que se requiere la implementación de una etapa de preprocesamiento robusta que permita eliminar el ruido proveniente del sensor antes de determinar la orientación del vehículo.

### 2.1.1. Protocolo de adquisición

Para construir la base de datos se toman las mediciones correspondientes a la aceleración y giro en los 3 ejes del sensor acoplado a la placa principal del robot controlado en pleno vuelo. La placa principal del vehículo está compuesta por un micro-controlador Atmel ATMEGA2560, un módulo Bluetooth HC-05, el sensor de medición de fuerzas inerciales GY-521 y un shield driver L293D para el control de los 4 motores de DC de 9 voltios acoplados a las hélices del quadcopter. Es importante resaltar que la utilización del módulo de comunicación por Bluetooth se incluye con el propósito de enviar la información proveniente del sensor a una unidad de procesamiento independiente con respecto al dispositivo que manipula las variables de control del vehículo. Debido a que el controlador del UAV cuenta con capacidades limitadas de procesamiento, con esto se garantiza que no se afecta el tiempo de respuesta del quadcopter ni los tiempos requeridos para el procesamiento de las variables controladas del vehículo durante el tiempo de vuelo para evitar accidentes con este motivo. Entonces, el controlador se configura de tal manera que envía los datos provenientes del sensor a una estación en tierra en la cual se procesan y almacenan. Sin embargo, a pesar de que esto representa una limitación en la implementación sobre sistemas reales, también se evaluará la capacidad requerida en tiempo y memoria del algoritmo de aprendizaje para su implementación del modelo ajustado sobre la plataforma principal del UAV. Para la adquisición de los datos, se lleva el vehículo a una altura de 1 metro sobre el suelo en primera instancia; y se mantiene en una posición completamente horizontal ( $0^\circ$  de inclinación). Seguido, se incrementa a velocidad máxima los motores del vehículo para tomar las muestras del dispositivo. La medición inicia 2 segundos después de comenzar la elevación del vehículo, con el fin de permitir que los motores alcancen su velocidad máxima en el momento de realizar la adquisición. A partir de allí, se toman los datos por un periodo de 5 segundos, lo cual permite extraer aproximadamente 511 muestras para el vehículo en esta inclinación. El mismo procedimiento es realizado para cada una de las inclinaciones mostradas en la Figura 2, en las cuales se considera un ángulo de  $45^\circ \pm 5^\circ$  para cada caso. En total se adquieren 4599 muestras correspondientes a todas las inclinaciones.

### 2.1.2. Preprocesamiento usando Filtro Kalman

Dada la naturaleza de las mediciones y con el fin de incrementar la precisión de los datos en lapsos de tiempos cortos, se incluye la utilización de una etapa de filtrado que involucra un

filtro de Kalman. El objetivo del filtro es identificar y eliminar ruido blanco, suavizar la señal obtenida y mejorar la precisión de la medición en el sistema. Debido a que el filtro de Kalman utiliza únicamente la medición del sensor y su matriz de incertidumbre, este pudo aplicarse en tiempo real sobre la estación en tierra durante el momento de la adquisición de los datos.

Un ejemplo de los datos obtenidos de la base de datos, una vez aplicado el filtro Kalman se muestra en la Tabla 1; en la cual se toman muestras aleatorias de cada tipo de inclinación. Estas muestras son consideradas como entradas para los algoritmos utilizados en la etapa de clasificación, en la cual se identificará la inclinación en el vehículo.

## 2.2. Etapa de análisis de datos y estrategias de aprendizaje

Considerando la necesidad de discriminar entre diferentes tipos de inclinación, se hace necesario realizar una búsqueda de diferentes métodos que permitan la separación de múltiples clases (MacQueen, 1995). En este sentido, se propone la utilización de estrategias ampliamente utilizadas en la literatura para resolver problemas de datos de múltiple etiqueta, tales como la clasificación Bayesiana (Rish, 1999), de K vecinos cercanos (k-NN) (Cover and Hart, 1967) y máquinas de vectores de soporte (SVM) (Scholkopf, 2006). Con el fin de analizar los resultados obtenidos por los modelos, cada estrategia es evaluada usando una validación cruzada (Chapman and Sonnenberg, 1995), de la cual se extraen métricas como precisión, tiempo de procesamiento y matriz de confusión. Todas estas estrategias son implementadas sobre el software Matlab®.

### 2.2.1. Clasificador de Bayes

El clasificador de Bayes establece una hipótesis sobre la clase a la cual pertenece una muestra ( $x_j$ ) que contiene los datos de aceleración y giro del subconjunto total de datos obtenidos ( $X$ ) a un grupo ( $y_j$ ) correspondiente a la inclinación del subconjunto de tipos de inclinaciones ( $Y$ ). En este se considera la probabilidad ( $P(Y)$ ) de cada clase y se maximiza mediante (1). La regla con la cual se estima la probabilidad de que una muestra pertenezca a un subconjunto de datos se describe en (2). El clasificador de Bayes se describe completamente en (Castrillón et al., 2008).

$$\text{Arg}_Y [\text{Max}P(Y|X)] \quad (1)$$

$$P(Y|X) = \frac{P(Y) * P(X|Y)}{P(X)} \quad (2)$$

El problema de clasificación de (2) basado en (1) puede expresarse en (3).

$$\text{Arg}_Y \left[ \frac{\text{Max}(P(Y)P(X|Y))}{P(X)} \right] \quad (3)$$

### 2.2.2. K vecinos cercanos (k-NN)

Mediante el clasificador no paramétrica, k-NN se permite estimar la probabilidad de que el subconjunto de datos de aceleración  $X_j$  pertenezca a un tipo de inclinación  $Y_j$  mediante la observación del valor de la función de densidad de probabilidad. Su característica principal en la clasificación de las clases está basada en la identificación de  $k$  ejemplos de entrenamiento cercanos a la muestra en el espacio de características (Pernkopf,

Tabla 1: Muestras aleatorias tomadas de la base de datos en grados (deg)

Clase	Aceleración X	Aceleración Y	Aceleración Z	Giro X	Giro Y	Giro Z
Plano	-22.98	40.36	1.09	-0.2	0.1	-0.1
Adelante	0.49	-86.66	16.20	-0.1	0.1	-0.1
Atrás	2.04	86.96	20.58	-0.0	0.1	-0.1
Derecha	-87.64	1.46	26.72	0.3	-0.2	-0.0
Izquierda	87.50	1.76	26.82	-0.1	0.0	-0.1
Izq-Adel	34.77	-55.1	15.89	-0.1	0.3	-0.1
Der-Adel	-53.99	-35.9	17.82	-0.2	0.2	-0.2
Der-Atras	-43.32	46.5	15.74	-0.3	0.1	-0.3
Izq-Atras	35.88	54.0	19.02	-0.2	0.3	-0.2

2005). Con el fin de estimar aquellos ejemplos  $X_i$  que están cercanos a la muestra  $X_j$  se utiliza la función de distancia euclidiana, la cual se expresa en (4).

$$d(X_i, X_j) = \sqrt{\sum_{r=1}^P (X_{r*i} - X_{r*j})^2} \quad (4)$$

En la fase de clasificación, se calculan las distancias Euclídeas entre los  $k$  vectores de entrenamiento (subconjunto de datos obtenidos  $X_i$  con un tipo de inclinación  $Y_i$ ) con respecto a los de la muestra desconocida. El valor del ejemplo de clasificación será estimado a través de una ponderación de las inclinaciones de los datos más cercanas a la muestra que se desea clasificar. Con el fin de determinar de manera óptima el número de vecinos  $k$  del modelo, se realiza una experimentación en la cual se evalúa el modelo con diferentes valores de  $k$  en un rango seleccionado heurísticamente ( $k_{min} = 2$ ,  $k_{max} = 20$ ). A partir de esto, se selecciona el valor que obtiene la mayor tasa de desempeño en términos de precisión  $k = 5$ .

### 2.2.3. Máquinas de soporte de vectores (SVM)

El objetivo de sintonizar una máquina de soporte de vectores consiste en encontrar el hiperplano óptimo que permita separar los tipos de inclinación asociados al vehículo. Encontrar el hiperplano óptimo significa que el margen de separabilidad entre las inclinaciones debe ser el más amplio con el fin de garantizar equitatividad en la clasificación de las muestras y reducir el margen de error esperado. La ecuación característica del hiperplano se representa en (5).

$$F(X) = X' * \beta + b = 0 \quad (5)$$

Donde  $\beta$  y  $b \in \mathbf{R}$  y  $X$  es un subconjunto de la base de datos  $x_j$  con sus correspondientes inclinaciones  $y_j$  en una dimensión  $d$ . El problema principal del hiperplano consiste en encontrar a  $\beta$  y  $b$  de tal forma que se permita minimizar  $\|\beta\|$  para todo el grupo de datos  $(x_j, y_j)$ . El vector de soporte de  $x_j$  es el óptimo cuando se cumple la regla presentada en (6).

$$Y_j * F(X_j) = 1 \quad (6)$$

Debido a la no trivialidad de los datos de entrada para la SVM, se realiza la implementación de la función kernel de base radial (RBF), con el fin de establecer mayor separabilidad entre las clases. El kernel RBF realiza un mapeo entre cada subconjunto de  $(x_j, x_i)$  a un espacio lineal denominado  $S$  mediante una

función  $\varphi$ . La función de separación RBF para una muestra  $\varphi$  es descrita en (7).

$$G(X_j, X_i) = \exp\left(\frac{-(X_j - X_i) * (X_j - X_i)}{2\varphi^2}\right) \quad (7)$$

En donde  $(x_j - y_j)$  es reconocida como la distancia euclídea entre dos muestras o vectores de características.

### 2.3. Evaluación de las técnicas de aprendizaje

Con el fin de evaluar las estrategias de clasificación a partir de los datos descritos en la (sección 2.1) se realiza una subdivisión de los datos por grupos usando el algoritmo de validación cruzada para 10 grupos (Chapman and Sonnenberg, 1995). Se selecciona esa cantidad estratégicamente, debido a su relevancia estadística en comparación con las evaluaciones usando otras cantidades de subconjuntos. Seguido, se entrenan los modelos y una vez ajustados, se utilizan los datos de prueba para obtener matrices de confusión por cada subconjunto. Adicionalmente, considerando la implementación de estos modelos computacionales sobre una plataforma principal de un UAV; una vez ajustados los modelos utilizando los datos de entrenamiento se mide el tiempo tomado por el algoritmo para clasificar de manera individual cada una de las instancias de prueba. Las estrategia de validación cruzada y las métricas extraídas se describen claramente en las subsecciones 2.3.1 y 2.3.2.

#### 2.3.1. Validación cruzada

Esta estrategia de validación de resultados en la literatura es conocida debido a su relevancia estadística para la fiabilidad en la evaluación de sistemas de aprendizaje computacional. Consiste en dividir el conjunto de datos totales en subgrupos más pequeños, estratificados estadísticamente con los que los modelos usarán algunos para validación y los demás para entrenamiento. En el caso de este trabajo, se divide el conjunto total en 10 subconjuntos en los que, cada iteración se utilizarán 9 para entrenamiento y uno para validación. Las métricas consideradas a continuación (precisión y matrices de confusión) son extraídas a partir de la evaluación de los modelos sobre estos subconjuntos. Adicionalmente, una vez los modelos son entrenados se reporta también los tiempos requeridos para la predicción de 1 muestra por cada modelo computacional. Estos tiempos son medidos usando una computadora de propósito general que incluye una CPU Intel® Core™ i7 (8 threads) de 2.6GHz. En la figura 4 se muestra una representación gráfica de las distribuciones mediante las cuales se extrae el modelo.

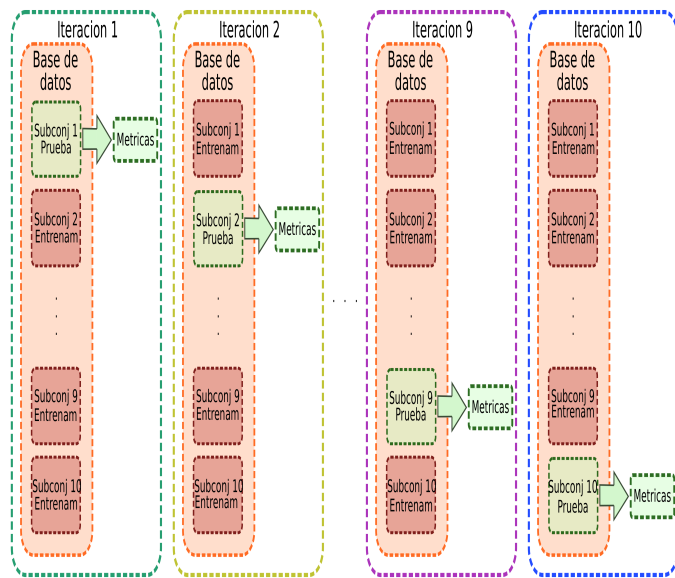


Figura 4: Distribución de las muestras de entrenamiento y prueba para cada iteración usando validación cruzada.

### 2.3.2. Matrices de confusión y precisión

La precisión estimada de los modelos para un subconjunto de datos de prueba permite estimar la cantidad de muestras determinadas como positivos aciertos con respecto a la cantidad de instancias determinadas de manera incorrecta. Esta medición, sobre conjuntos de datos balanceados; permite determinar el error obtenido por cada uno de los modelos al momento de predecir instancias no vistas después de el entrenamiento. Con esta medida, se determina cuán efectivo el modelo se comporta ante diversos comportamientos en las muestras. La precisión para cada subconjunto es obtenida a partir de la ecuación 8, y el promedio de todas las precisiones obtenidas se reporta en este trabajo.

Adicionalmente, se reporta además la matriz de confusión, en la cual se describen la cantidad de muestras y las clases con las cuales se “confunde” el modelo al realizar predicciones. Esta métrica es útil dado que permite identificar aquellos grupos de muestras en los cuales el modelo posee comportamientos confusos con respecto a ciertas muestras. En este sentido, la matriz de confusión se calcula teniendo en cuenta las predicciones realizadas por el modelo para un conjunto de  $X$  muestras y las etiquetas reales del mismo conjunto. A partir de allí, se ubican la cantidad de muestras predichas correctamente para la clase actual (verdaderos positivos), la cantidad de muestras predichas para las demás clases (verdaderos negativos), la cantidad de muestras de otras clases predichas como la clase actual (falsos negativos) y las muestras de la clase actual predichas como muestras de otras clases (falsos positivos). Estas métricas permiten establecer con mayor claridad aquellas clases en las cuales se presenta mayor confusión, ya sea por similitud o ineficiencia en la separación de clases por los modelos. En la figura 5 se muestra la distribución de los índices mencionados para una matriz de confusión de múltiples clases.

$$\text{precisión} = \frac{\text{predicciones correctas}}{\text{total instancias}} \quad (8)$$

		Predicciones					Predicciones							
		C1	C2	...	Ck	...	Cn-1	Cn	C1	...	Ck	...	Cn-1	Cn
Etiquetas	C1	Verdaderos Negativos	Falsos Positivos		Falsos Negativos		Verdaderos Negativos	Verdaderos Positivos	Falsos Negativos	Falsos Positivos	Verdaderos Negativos			
	C2													
	Ck				Verdaderos Positivos									
	Cn-1													
	Cn	Verdaderos Negativos	Falsos Positivos		Falsos Negativos		Verdaderos Negativos	Verdaderos Positivos	Falsos Negativos	Falsos Positivos	Verdaderos Negativos			

Figura 5: Interpretación de la matriz de confusión para la clase  $C_k$  de un problema de  $C_n$  clases.

## 3. Resultados

En esta sección se presentan los resultados más relevantes obtenidos a partir de las experimentaciones planteadas.

### 3.1. Clasificación usando Bayes, $k$ -NN y SVM

Con el fin de realizar una clasificación efectiva y confiable de la inclinación del UAV, la implementación de cada uno de los algoritmos de entrenamiento se realiza usando validación cruzada, usando 10 subconjuntos dada su significancia estadística (Chapman and Sonnenberg, 1995). Adicionalmente, se considera que los datos estén estratificados estadísticamente, con el fin de evitar sesgos producidos por desbalanceo entre muestras durante el ajuste de parámetros. Los resultados de cada subconjunto son evaluados a través de matrices de confusión y se reporta el resultado promedio obtenido por ellas. Adicionalmente, debido a la consideración de implementación del sistema sobre la placa de un robot aéreo (sistema embebido de recursos computacionales limitados) se evalúa también el tiempo de procesamiento para cada iteración en etapa de validación y entrenamiento. En este sentido, el valor reportado fue medido teniendo en cuenta el tiempo promedio requerido por el algoritmo una vez ajustado para clasificar las instancia perteneciente a cualquier inclinación en etapa de validación. Utilizando validación cruzada al implementar el clasificador SVM con kernel de base radial se obtienen los resultados mostrados en la figura 6. Allí es posible evidenciar que los resultados obtenidos por el clasificador poseen 0,998 de efectividad en la predicción de las muestras, lo cual lo hace el sistema más efectivo con respecto a los demás de clasificadores. Sin embargo, a pesar de su efectividad se evidencia en la tabla 2 un tiempo computacional considerable para clasificar una instancia al finalizar el entrenamiento. Por otra parte, en la figura 8 se muestran los resultados obtenidos por el clasificador  $k$ -NN; el cual posee una notable tasa de reconocimiento de 0,969, a pesar de no alcanzar los obtenidos por el SVM. Sin embargo, el tiempo de procesamiento es significativamente inferior en comparación con el obtenido por la SVM. Esto hace más apropiada una implementación sobre una plataforma de recursos limitados sin una reducción significativa en la clasificación de las instancias.

**Confusion Matrix**

Inclinaciones	Adelante	51 11.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	Atras	0 0.0%	50 10.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	98.0% 2.0%	
	Derecha	0 0.0%	0 0.0%	51 11.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	Izquierda	0 0.0%	0 0.0%	0 0.0%	51 11.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	Der - Adel	0 0.0%	0 0.0%	0 0.0%	0 0.0%	51 11.1%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	Der - Atras	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	51 11.1%	0 0.0%	0 0.0%	100% 0.0%	
	Izq - Adel	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	51 11.1%	0 0.0%	100% 0.0%	
	Izq - Atras	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	51 11.1%	100% 0.0%	
	Neutral	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	51 11.1%	100% 0.0%
		100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	98.1% 1.9%	100% 0.0%	99.8% 0.2%
	Ad	At	D	I	D - Ad	D - At	I - Ad	I - At	N		
	<b>Predicciones</b>										

Figura 6: Resultados algoritmo SVM

**Confusion Matrix**

Inclinaciones	Adelante	47 10.2%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	1 0.2%	2 0.4%	0 0.0%	0 0.0%	92.2% 7.8%
	Atras	0 0.0%	50 10.9%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	98.0% 2.0%
	Derecha	0 0.0%	1 0.2%	48 10.5%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	1 0.2%	94.1% 5.9%
	Izquierda	0 0.0%	0 0.0%	1 0.2%	48 10.5%	1 0.2%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	94.1% 5.9%
	Der - Adel	1 0.2%	0 0.0%	0 0.0%	0 0.0%	49 10.7%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	96.1% 3.9%
	Der - Atras	1 0.2%	0 0.0%	1 0.2%	2 0.4%	0 0.0%	47 10.2%	0 0.0%	0 0.0%	0 0.0%	92.2% 7.8%
	Izq - Adel	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.4%	47 10.2%	1 0.2%	0 0.0%	92.2% 7.8%
	Izq - Atras	0 0.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	49 10.7%	0 0.0%	96.1% 3.9%
	Neutral	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 10.9%	98.0% 2.0%
		92.2% 7.8%	98.0% 2.0%	94.1% 5.9%	94.1% 5.9%	96.1% 3.9%	92.2% 7.8%	92.2% 7.8%	96.1% 3.9%	98.0% 2.0%	94.8% 5.2%
	Ad	At	I	D	D - Ad	D - At	I - Ad	I - At	N		
	<b>Predicciones</b>										

Figura 7: Resultados algoritmo Bayes

**Confusion Matrix**

Inclinaciones	Adelante	51 11.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	Atras	0 0.0%	47 10.2%	0 0.0%	1 0.2%	1 0.2%	1 0.2%	1 0.2%	0 0.0%	0 0.0%	92.2% 7.8%
	Derecha	0 0.0%	1 0.2%	49 10.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	96.1% 3.9%
	Izquierda	0 0.0%	1 0.2%	0 0.0%	49 10.7%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	96.1% 3.9%
	Der - Adel	0 0.0%	1 0.2%	0 0.0%	1 0.2%	49 10.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	96.1% 3.9%
	Der - Atras	0 0.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	50 10.9%	0 0.0%	0 0.0%	0 0.0%	98.0% 2.0%
	Izq - Adel	0 0.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	50 10.9%	0 0.0%	0 0.0%	98.0% 2.0%
	Izq - Atras	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	50 10.9%	0 0.0%	98.0% 2.0%
	Neutral	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 10.9%	98.0% 2.0%
		100% 0.0%	92.2% 7.8%	96.1% 3.9%	96.1% 3.9%	96.1% 3.9%	98.0% 2.0%	98.0% 2.0%	98.0% 2.0%	98.0% 2.0%	96.9% 3.1%
	Ad	At	D	I	D - Ad	D - At	I - Ad	I - At	N		
	<b>Predicciones</b>										

Figura 8: Resultados algoritmo k-NN

Tabla 2: Tiempos de procesamiento

Algoritmo	Bayes	k-NN	SVM
Tiempo (s)	0.0013	0.0755	1.6158

Por ultimo, el algoritmo de Bayes obtiene los resultados reportados en la figura 7 con una tasa de reconocimiento de 0,948. A pesar de que este clasificador posee la menor tasa de reconocimiento entre clasificadores, obtiene el menor tiempo de procesamiento. Esto debido a la regla empleada para realizar la estimación de las instancias la cual no requiere gran complejidad matemática para resolverse. Este algoritmo muestra resultados prometedores para su implementación en sistemas reales no pilotados de plataformas de bajo costo y limitadas prestaciones. Por otra parte, es importante resaltar que los resultados obtenidos por los modelos descritos anteriormente se realiza sin tener en cuenta sus parámetros dinámicos; como se sugiere al utilizar técnicas de control modernas avanzadas las cuales varían con respecto al vehículo utilizado (Kumagai and Ochiai, 2008; Carabin et al., 2017). A partir de este trabajo en el cual se realizan estrategias basadas en aprendizaje computacional se permite la identificación de la inclinación de un vehículo sin tener en cuenta la variación de los parámetros dinámicos particulares entre cada vehículo aéreo. Esta identificación requiere únicamente la utilización de los datos adecuadamente filtrados de los sensores inerciales. De esta manera se permitiría la implementación de modelos que no necesariamente consideren las variaciones dinámicas entre vehículos con el fin de alcanzar el desarrollo de algoritmos generalizados para su implementación en diversas plataformas móviles. Adicionalmente, estas técnicas podrían hacerse extensibles a otro tipo de modelos robóticos (e.g. robots terrestres) dado que la estimación de la inclinación en los mismos se realiza de manera similar en el desarrollo de sistemas de navegación u orientación. Así entonces, se evidencia que el uso de estrategias basadas en aprendizaje computacional podría prometer importantes contribuciones en áreas relacionadas con la robótica móvil teniendo en cuenta su implementación en sistemas de bajo costo y limitadas capacidades de procesamiento.

#### 4. Conclusiones

En este trabajo se presenta la utilización de distintos algoritmos de aprendizaje computacional con el fin de determinar la inclinación de un vehículo aéreo no tripulado (UAV) para brindar orientación en tiempo de vuelo del cual se desconoce el modelo cinemático. Los resultados presentados por los tres técnicas usadas demuestran gran efectividad al lograr las tasas de acierto notables en el problema de identificación de la inclinación del vehículo.

Para resolver esto, se propone un experimento con el fin de obtener los datos provenientes de un sensor inercial (IMU) correspondientes a aceleración y velocidad angular sobre el mismo durante tiempo real de vuelo. Los datos de los sensores son filtrados usando el algoritmo de Kalman previo a la etapa de reconocimiento para maximizar la efectividad de los clasificadores.

Para el reconocimiento se utilizan los clasificadores de máquinas de soporte vectorial (SVM),  $k$  vecinos cercanos (k-NN) y Bayes. Los resultados obtenidos por los algoritmos obtienen resultados prometedores que varían entre 0,95 y 0,99 para la tarea de identificación de la inclinación. Adicionalmente se reportan los tiempos promedio de clasificación de cada instancia en los algoritmos mencionados anteriormente, una vez ajustados los parámetros de los modelos.

Es importante resaltar que la estrategia descrita en este trabajo requiere únicamente de la información tomada y filtrada a partir de los sensores inerciales de bajo costo, sin tener en cuenta las variaciones dinámicas particulares del vehículo aéreo como es sugerido al utilizar técnicas de control modernas avanzadas en el estado del arte. Esto sugeriría el desarrollo de estrategias generalizadas que permitan el control de vehículos no tripulados sin tener en cuenta las variaciones específicas entre dos vehículos distintos a pesar de sus diversas variaciones físicas. Adicionalmente, la estrategia propuesta en este trabajo podría hacerse extensible a otro tipo de robots dado que la estimación de la inclinación se realiza de manera similar en comparación al trabajo propuesto.

Por último, se propone como trabajos futuros la utilización de este modelo en un UAV en tiempo real, que utilice una plataforma de recursos computacionales limitados. Esto, con el fin de eliminar la limitante del sistema dependiente de la estación de procesamiento en tierra, además de que permitirá validar las estrategias planteadas para la implementación sobre la plataforma principal de un UAV en pleno vuelo. Adicionalmente, se propone la evaluación de los modelos expuestos sobre robots vertebrados terrestres para el desarrollo de un sistema de navegación más eficaz.

## Referencias

- Carabin, G., Vidoni, R., Mazzetto, F., Gaspardo, A., 2017. Dynamic model and instability evaluation of an articulated mobile agri-robot. In: *Advances in Italian Mechanism Science*. Springer, pp. 335–343.
- Castrillón, O. D., Giraldo, J. A., C, W. A. S., 2008. Sistema de Clasificación Bayesiano basado en Múltiples Clases. *Sistemas, Cibernética e Informática* 5 (1), 25–28.
- Chapman, G. B., Sonnenberg, F. A., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *International joint conference on artificial intelligence (IJCAD)*. p. 456.  
URL: <http://books.google.es/books?id=c0XGM6mIMakC>  
DOI: 10.1067/mod.2000.109031
- Colmenares-Quintero, R. F., Góez-Sánchez, G. D., 2018. Route planning in real time for short-range aircraft with a constant-volume-combustor-geared turboprop to minimize operating costs by particle swarm optimization. *Cogent Engineering* 5 (1).  
URL: <https://doi.org/10.1080/23311916.2018.1429984>  
DOI: 10.1080/23311916.2018.1429984
- Cover, T. M., Hart, P. E., 1967. Nearest Neighbor pattern classification. *Information Theory, IEEE Transactions on* 1 (13), 21–27.  
DOI: 10.1007/978-0-387-30164-8
- Darío, R., Tarazona, F., Lopera, F. R., 2014. Anti-collision System for Navigation Inside an UAV Using Fuzzy Controllers and Range Sensors. 978-1-4799-7666-9/14/\$31.00 ©2014 IEEE 2014 XIX Symposium on Image, Signal Processing and Artificial Vision (STSIVA).  
DOI: 10.1109/STSIVA.2014.7010153
- Giusti, A., Guzzi, J., Ciresan, D. C., He, F.-L., Rodríguez, J. P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Di Caro, G., et al., 2016. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters* 1 (2), 661–667.
- Góez, G.-D., 2016. Planeamiento de rutas en vehículos aéreos no tripulados usando algoritmos bio-inspirados sobre sistemas. Ph.D. thesis, Instituto Tecnológico Metropolitano.
- Góez, G. D., Velásquez Velez, R. A., Botero Valencia, J. S., 2016. On-Line Route Planning of Uav Using Particle Swarm Optimization on Microcontrollers. *IEEE Latin America Transactions* 14 (4), 1705–1710.  
DOI: 10.1109/TLA.2016.7483504
- Goslinski, J., Giernacki, W., Gardecki, S., 2013. Unscented Kalman Filter for an orientation module of a quadrotor mathematical model. 2013 9th Asian Control Conference, ASCC 2013.  
DOI: 10.1109/ASCC.2013.6606269
- Kan, E. M., Lim, M. H., Ong, Y. S., Tan, A. H., Yeo, S. P., 2013. Extreme learning machine terrain-based navigation for unmanned aerial vehicles. *Neural Computing and Applications* 22 (3-4), 469–477.
- Kumagai, M., Ochiai, T., 2008. Development of a robot balancing on a ball. In: 2008 International Conference on Control, Automation and Systems, IC-CAS 2008. pp. 433–438.  
DOI: 10.1109/ICCAS.2008.4694680
- Li, R., Liu, J., Zhang, L., Hang, Y., Sept 2014. Lidar/mems imu integrated navigation (slam) method for a small uav in indoor environments. In: 2014 DGON Inertial Sensors and Systems (ISS). pp. 1–15.  
DOI: 10.1109/InertialSensors.2014.7049479
- Lupashin, S., 2011. Quadcopter Ball Juggling. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 5113–5120.  
URL: <file:///home/youssef/Documents/06094506.pdf>
- MacQueen, J., 1995. Some methods for classification and analysis of multivariate observations. *University of California, Los Angeles* 199 (233), 281–296.  
DOI: citeulike-article-id:6083430
- Pernkopf, F., 2005. Bayesian network classifiers versus selective k-NN classifier. *Pattern Recognition* 38 (1), 1–10.  
DOI: 10.1016/j.patcog.2004.05.012
- Pieters, P., 2009. Versatile {MEMS} and mems integration technology platforms for cost effective {MEMS} development. *Microelectronics and Packaging Conference, 2009. {EMPC} 2009*. European, 1–5.
- Puls, T., Hein, A., 2010. 3D trajectory control for quadcopter. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 640–645.  
DOI: 10.1109/IROS.2010.5650249
- Razak, N. A., Arshad, N. H. M., Adnan, R., Misnan, M. F., Thamrin, N. M., Mahmud, S. F., 2013. A study of Kalman's filter in embedded controller for real-time quadcopter roll and pitch measurement. *Proceedings - 2012 IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2012*, 590–595.  
DOI: 10.1109/ICCSCE.2012.6487214
- Rish, I., 1999. An empirical study of the naive Bayes classifier. *T.J. Watson Research Center*, 41–46.  
DOI: 10.1.1.330.2788
- Scholkopf, B., 2006. Smola, Learning with kernels : support vector machines, regularization, optimization, and beyond. No. February.  
DOI: 10.1017/CB09781107415324.004
- Song, Y., Meng, Q. H., Luo, B., Zeng, M., Ma, S. G., Qi, P. F., 2016. A wind estimation method for quadrotors using inertial measurement units. 2016 IEEE International Conference on Robotics and Biomimetics, ROBIO 2016 (2), 426–431.  
DOI: 10.1109/ROBIO.2016.7866359
- Tarazona, R. D. F., Lopera, F. R., Sánchez, G. D. G., 2015. Anti-collision system for navigation inside an UAV using fuzzy controllers and range sensors. 2014 19th Symposium on Image, Signal Processing and Artificial Vision, STSIVA 2014.  
DOI: 10.1109/STSIVA.2014.7010153
- Von Chong, A., Caballero, R., 2014. Adaptive Kalman filtering for the estimation of orientation and displacements in submarine systems. *Proceedings of the 2014 IEEE Central America and Panama Convention, CONCAPAN 2014 (Concapan Xxxiv)*.  
DOI: 10.1109/CONCAPAN.2014.7000439
- Wendel, J., Meister, O., Schlaile, C., Trommer, G. F., 2006. An integrated gps/mems-imu navigation system for an autonomous helicopter. *Aerospace Science and Technology* 10 (6), 527–533.
- Zul Azfar, A., Hazry, D., 2011. A simple approach on implementing IMU sensor fusion in PID controller for stabilizing quadrotor flight control. *Proceedings - 2011 IEEE 7th International Colloquium on Signal Processing and Its Applications, CSPA 2011*, 28–32.  
DOI: 10.1109/CSPA.2011.5759837