

Trabajo de Fin de Grado Ingeniería Aeroespacial

Estudio de integración de sensores en UAVs

Autor: José Luis Carretero Rodríguez

Tutora: Juana María Martínez Heredia

**Departamento de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2016



Trabajo de Fin de Grado
Ingeniería Aeroespacial

Estudio de integración de sensores en UAVs

Autor:

José Luis Carretero Rodríguez

Tutora:

Juana María Martínez Heredia

Departamento de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2016

Trabajo de Fin de Grado: Estudio de integración de sensores en UAVs

Autor: José Luis Carretero Rodríguez
Tutora: Juana María Martínez Heredia

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Índice general

1. Introducción	4
2. Modelado en el Espacio de Estados	7
2.1. Representación de sistemas en el espacio estados	8
2.2. Obtención de la representación en espacio de estados de sistemas discretos	9
2.2.1. Método de programación directa	10
2.2.2. Método de programación anidada	12
2.3. Relación entre la representación en espacio de estados y la función de transferencia . .	13
2.4. No unicidad de la representación en espacio de estados de un sistema	15
2.5. Resolución de las ecuaciones del espacio de estados	15
2.5.1. Procedimiento recursivo	16
2.5.2. Matriz de transición de estados	16
2.5.3. Método basado en la transformada Z	17
2.6. Linealización de las ecuaciones de estado	17
2.6.1. Interpretación Gráfica	18
2.6.2. Aproximación lineal del modelo	19
2.7. Discretización de las ecuaciones de estado continuas	20
2.8. Controlabilidad y Observabilidad	23
2.8.1. Controlabilidad del estado completo	25
2.8.2. Controlabilidad de la salida	26
2.8.3. Observabilidad	27
2.8.4. Principio de Dualidad	28
2.9. Transformación de un sistema en formas canónicas	29
2.9.1. Forma Canónica Controlable	29
2.9.2. Forma Canónica Observable	30
2.10. Descripción de un sistema en parte controlable/observable y no controlable/no observable	31
2.10.1. Parte controlable/no controlable	31
2.10.2. Parte observable/no observable	32
3. Modelado de Sensores en los Vehículos Aéreos no Tripulados	34
3.1. Caracterización de los Sensores	35

3.1.1. Descriptores estáticos	35
3.1.2. Descriptores dinámicos	37
3.2. Uso correcto de los sensores	38
3.2.1. Calibración de sensores	38
3.2.2. Modelado de simulaciones SIL-HIL - <i>Software-in-the-loop</i> y <i>Hardware-in-the-loop</i>	39
3.3. Sensores para medir distancias y proximidad	40
3.3.1. Sensores capacitivos	41
3.3.2. Sensores inductivos	42
3.3.3. Basados en efecto Hall	43
3.3.4. Basados en ultrasonidos	44
3.3.5. Sensores de Espectro Infrarrojo	45
3.3.6. Visión artificial	45
3.4. Sensores de luz	47
3.5. Sensores de Velocidad	48
3.5.1. Tubo de Pitot	50
3.6. Codificadores (<i>Encoders</i>)	51
3.7. Altímetros	52
3.8. Giroscopios, Acelerómetros y Magnetómetros: Los pilares de una IMU (del inglés, <i>Inertial Measurement Unit</i> - Unidad Inercial de Medida)	52
3.9. GPS - Global Positioning System	55
3.9.1. Segmentos	57
3.9.2. Disponibilidad, Integridad y Continuidad	58
4. Filtro de Kalman	60
4.1. Introducción	60
4.2. ¿Qué es el filtro de Kalman?	60
4.3. El modelo multidimensional	63
5. Caso práctico - Simulación en Matlab del Filtro de Kalman	68
5.1. Procesamiento de datos	68
5.1.1. Medida de la trayectoria - GPS	68
5.1.2. Medida de la trayectoria - Google Maps	70
5.2. Parámetros del filtro de Kalman	71
5.3. Resultados obtenidos de la simulación	72
6. Conclusiones y Líneas Futuras	76
A. Varianza & Covarianza	78
B. Código Matlab de caso práctico	79

Índice de figuras

1.1. Estabilidad y maniobrabilidad	5
2.1. Diagrama de bloques de la representación en espacio estados de un sistema LTI [13] . .	9
2.2. Ejemplo de linealización de una curva en un punto	19
2.3. Muestreo de un sistema en tiempo continuo	20
2.4. Ejemplificación del Teorema de Muestreo o Teorema de Nyquist	21
2.5. a) Sistema de Control con realimentación del estado. b) Sistema de control con reali- mentación del estado y observador.	24
2.6. Descomposición parte controlable y no controlable	32
2.7. Descomposición parte observable y no observable	33
3.1. Respuesta no lineal de un sensor [22]	36
3.2. Tipos de no linealidades [22]	37
3.3. Descriptores dinámicos en la respuesta de un sensor [22]	38
3.4. Esquemas de sensores capacitivos [22]	41
3.5. Esquema de sensor capacitivo como detector de proximidad	42
3.6. Ejemplo de transformador diferencial [22]	42
3.7. Sensores capacitivos e inductivos reales (cortesía de SirioRed)	43
3.8. Efecto Hall [22]	43
3.9. Explicación del error producido debido al ángulo de incidencia	44
3.10. Sensores de efecto Hall (izquierda, Bernstein) y ultrasonidos (derecha, Arduino HCSR04 [9])	45
3.11. Sensores de infrarrojos, Sharp GP2D15)	45
3.12. Diferentes tipos de cámaras de visión artificial. U-Camera de Airelectronics, Hero 3 de GoPro, Tetracam ADC Light, Tau de FLIR (<i>Grupo Acre</i>)	47
3.13. Esquema fotorresistencia	47
3.14. Esquema fotodiodo	48
3.15. Esquema fototransistor	48
3.16. Tacogenerador: rueda dentada acciona el interruptor magnético [22]	49
3.17. Esquema del funcionamiento de un tacómetro. [22]	49
3.18. Tacómetro para UAVs y helicópteros de aeromodelismo, empresa SkyRC	50
3.19. Esquema del tubo de pitot-estática con un sensor de presión junto con uno real para UAVs de la empresa UAV Factory [3]	50
3.20. Ejemplos de encoders incremental, absoluto y código Gray	51

3.21. Altímetro barométrico BMP085 de Adafruits	52
3.22. Esquema del funcionamiento de un altímetro radiométrico	52
3.23. Esquema giroscopio y giroscopio MPU-6050 de 3 ejes compatible con Arduino, de empresa Oem	53
3.24. Medidas de acelerómetro y acelerómetro Arduino Adxl335 de 3 ejes	53
3.25. Magnetómetro de 3 ejes HMC5883L, de Naylamp Mechatronics	54
3.26. Módulo de Sensor IMU GY-85 de 9 ejes (Giróscopo, Acelerómetro y Magnetómetro), de Banggood	54
3.27. Medida de los pseudorangos de 4 satélites usados para triangular la posición del receptor [3]	56
3.28. Segmentos de Control de GPS	58
3.29. Ejemplo de DGPS basado en GBAS	59
4.1. Proceso iterativo del Filtro de Kalman	61
4.2. Explicación del uso de la Ganancia de Kalman	62
4.3. Proceso Iterativo que explica el modelo multidimensional mediante las ecuaciones del algoritmo	64
5.1. Ejemplo del formato del archivo procesado NMEA	69
5.2. Seguimiento de la trayectoria gracias al programa Google Earth. <i>Recorrido por la ciudad de Badajoz</i>	69
5.3. Trayectoria en línea recta y a velocidad constante. <i>Puente de la Universidad, Badajoz</i>	70
5.4. Coordenadas de los puntos inicial y final del puente según Google Maps	71
5.5. Comparación de las señales GPS, Google Maps y filtro de Kalman en altitud	73
5.6. Comparación de las señales GPS, Google Maps y filtro de Kalman en altitud, P más baja	74
5.7. Comparación de las señales GPS, Google Maps y filtro de Kalman en ejes (x,y)	75

1 | Introducción

En las últimas décadas se ha producido un avance realmente significativo en cuanto al desarrollo de los sistemas que integran los vehículos aéreos no tripulados, UAVs (del inglés, *Unmanned Aerial Vehicle*) o RPAS (del inglés, *Remotely Piloted Aircraft Systems*). Debido al gran rango de aplicaciones para las cuales este tipo de vehículos está capacitado, se produjo un salto de su uso exclusivo en el ámbito militar al ámbito civil, ampliando aún más su campo de actuación.

Es bien sabido que este tipo de aeronaves no tripuladas son utilizadas en tareas de búsqueda y rescate, exploración de edificios, terremotos, vigilancia comercial, control de incendios, salvamento marítimo, fotografía, vídeo y cartografía aérea, agricultura, entre otras muchas. La principal característica que ofrecen estos llamados *drones*, la cual los hace tan prácticos, es el gran potencial debido a su rápido y fácil acceso a zonas cuyo terreno es irregular, accidentado o simplemente peligroso, además de ofrecer imágenes aéreas a vista de pájaro y todo tipo de información procedente de los sensores que los integran.

El sector aeroespacial se ha visto en la obligación de investigar y desarrollar una nueva rama de esta disciplina dedicada al estudio de los UAVs, en gran parte debido a un gran aumento en su uso, puesto que son dispositivos al alcance casi de cualquier aficionado a la aviación. Puesto que no se dispone de un ser humano a bordo capaz de tomar las decisiones necesarias ante cualquier imprevisto, se requiere implementar una serie de dispositivos, tanto sensores como actuadores capaces de actuar conforme a cualquier situación que se les presente de la forma más similar a la que actuaría un miembro de la tripulación.

El gran impulso que han sufrido los vehículos no tripulados no se ha producido solamente gracias a los avances en el sector aeroespacial. No se concibe un dron sin los avances producidos en aviónica, electrónica, microcontroladores, sensores reducidos o MEMS (del inglés, *Micro Electromechanical Systems*), almacenamiento energético o control automático, entre otras. Con el fin de aumentar su fiabilidad, controlabilidad y autonomía, es necesario desarrollar tecnologías capaces de proporcionar al UAV una conciencia del estado en el que se encuentra y capacidad de adaptación.

El desarrollo de los sistemas de control de estas aeronaves es un proceso que debe tener en cuenta multitud de variables dinámicas, implícitas en los sistemas aerodinámicos. Normalmente estos sistemas son multivariables (deben tener en cuenta datos como posición, velocidad, rumbo, inclinación, distancia a objetos del entorno, etc.) y además presentan características no lineales (los sensores son dispositivos reales, y por tanto, tienen una serie de limitaciones en sus rangos de actuación tales como pueden ser la saturación, histéresis, zonas muertas, los cuales se explicarán más adelante). Por este motivo, es necesario un buen modelado en cuanto a la integración de las señales recibidas por dichos sensores. De esta forma se podrá dotar al sistema completo de un control suficiente para proporcionar al UAV un equilibrio entre **estabilidad** ante perturbaciones externas, y **maniobrabilidad** para un mayor acceso a áreas complicadas.

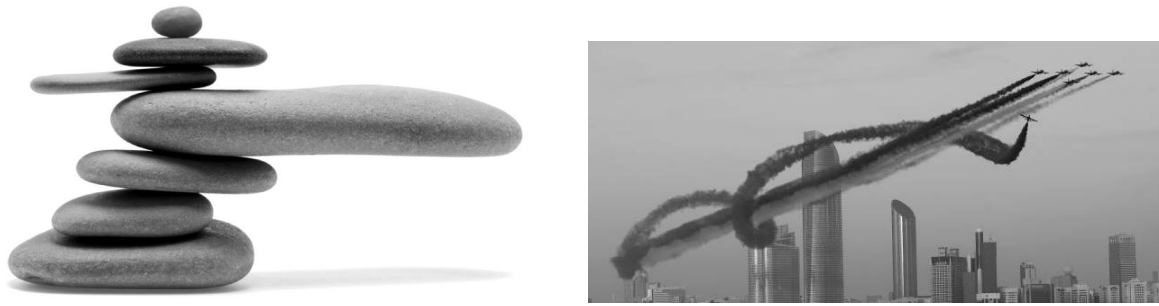


Figura 1.1: Estabilidad y maniobrabilidad

¿Quiénes son los encargados de proporcionar estas **dos características** a los dispositivos? Los sensores y los sistemas que lo integran. Para mantener una altura determinada es necesario en primer lugar conocer a qué altura se encuentra el vehículo: se necesita un altímetro. Para modificar la velocidad a la que se desplaza es necesario un sistema que, conociendo la velocidad actual, sea capaz de enviar una señal de control a los motores que mueven las hélices para cambiarla: el tubo de pitot o el GPS, por ejemplo, proporcionan la velocidad a la que se desplaza. Para utilizar el vehículo en piloto automático para llevarlo a una determinada posición, es necesario tanto evitar obstáculos de por medio (sensores de visión artificial, de distancia y proximidad), como saber de qué lugar parte (sensores de posición) y hacia donde se dirige (sensores de orientación como magnetómetros, giróscopos o acelerómetros).

En definitiva, una gran cantidad de sensores y sistemas los cuales se encuentran interconectados entre sí. Cada uno de ellos es capaz de medir una o varias variables, a veces varios de ellos proporcionan información sobre la misma variable. Es por ello que dichos sistemas se encuentran integrados, para obtener la mayor cantidad de información sobre un parámetro, de forma que si alguno de los sensores proporciona datos erróneos, tener otra fuente fiable en la que el sistema pueda basarse. Es por tanto necesario utilizar algún instrumento para trabajar con toda esta información recibida por los sensores, las cuales aportan datos sobre unas variables sobre las cuales es interesante conocer su valor (la posición del vehículo, la velocidad, la orientación, etc.)

Este es el motivo por el que se ha decidido comenzar este trabajo con un estudio sobre el **modelado del espacio estados**. El espacio estados es una herramienta de resolución de problemas muy útil a la hora de integrar diferentes variables (información recibida a partir de los distintos sensores y sistemas que integran un UAV) y datos de forma ordenada y precisa, como se verá en la primera parte de este trabajo. Es necesario conocer cómo los vehículos procesan la información que les proporcionan los diferentes tipos de sensores que lo conforman, para así ofrecer un modelo del sistema lo más real posible, y así cometer el menor número de errores posible.

Al ser una herramienta de resolución de problemas, ésta cuenta con una serie de desarrollos ya estudiados mediante los cuales se puede simplificar un problema complejo. Es decir, existen ya métodos de Resolución de las ecuaciones que conforman el problema del espacio estados, también procesos de linealización de dichas ecuaciones, discretización de ecuaciones continuas, y muchas otras características que se verán en el primer apartado de este tema. Como se ha dicho, se ha considerado oportuno incluir este apartado con un carácter más teórico para comprender mejor, llegado el momento, su aplicación en un problema real como es la simulación de un filtro de Kalman para una señal GPS. Se analizarán las características que posee el modelado de un sistema mediante el espacio estados, tales como la controlabilidad y la observabilidad de las variables con las que se trabaja, así como distintas maneras de representación de la misma, como son las diferentes formas canónicas.

Puesto que para obtener esas variables con las que va a trabajar el espacio estados se obtienen de la integración de sensores en un vehículo aéreo no tripulado, se ha dedicado a continuación un apartado al estudio de los **sensores** más comunes entre los drones. En dicho apartado se desglosan los diferentes tipos de dispositivos según su funcionalidad, explicando brevemente su funcionamiento, aplicaciones para las que es usado, errores que comete, etc. En dicho apartado se incluirán sensores que miden proximidad y distancia, velocidad, actitud, altitud, aceleración, etc. Se incluyen también algunas nociones a tener en cuenta a la hora de trabajar con los sensores, puesto que éstos poseen

errores y limitaciones de trabajo, y existen técnicas mediante las cuales se puede reducir el valor de dicho error, tales como son la correcta calibración del sensor o mediante modelados de simulaciones SIL-HIL (del inglés, *Software-in-the-loop*, *Hardware-in-the-loop*).

Pero los más destacables sistemas que se encuentran embarcados en un UAV son los **sistemas de navegación**. Éstos proporcionan al resto de subsistemas la posición, velocidad y actitud, información que necesitan para el control de la aeronave, gestionar una misión, o informar al piloto (en el caso de un UAV, el piloto es remoto). Desde los inicios de la navegación aérea se han utilizado **sistemas de navegación inerciales ó INS** (del inglés, *Inertial Navigation Systems*), los cuales poseían una baja precisión puesto que se acumulaban los errores en los cálculos. Con la mejora de dichos sistemas y la integración de éstos con los **GPS** (del inglés, *Global Positioning Systems*) y otros sensores, se puede obtener una navegación con un grado de precisión bastante alto. Existen numerosas actuaciones que realizan estas aeronaves no tripuladas gracias a la integración de sistemas INS/GPS.

El caso más típico entre los UAVs es la necesidad de integrar dos señales, una procedente del GPS y la otra procedente de un sistema de navegación inercial formado por acelerómetros, giróscopos y magnetómetros denominada IMU (del inglés, *Inertial Measurement Unit*). El algoritmo para llevar a cabo dicha integración más conocido es el **Filtro de Kalman**. Se trata de un proceso iterativo apoyado en una serie de ecuaciones matemáticas, las cuales proporcionan una solución que disminuye el error que existe en las dos señales recibidas de la IMU y del GPS. El filtro de Kalman se aplica a sistemas representados por el método del espacio estados, y sus principios de operación están divididos en dos partes: predicción y actualización, tal y como se verá en el apartado correspondiente. Dentro de los distintos posibles filtros que se pueden utilizar para este tipo de problemas el filtro de Kalman es el más atractivo puesto que es aquel que minimiza la varianza del error en la estimación.

Se analizarán las **bases teóricas** de dicho filtro y las ecuaciones en las que se basa siguiendo una serie de diagramas y procesos para facilitar su entendimiento. Se comenzará con una visión general de cómo funciona el filtro de Kalman para después explicar el modelo multidimensional más en detalle, en el cual se encuadraría el problema de la integración de la señal de una IMU y del GPS, y del resto de sensores que sean necesarios.

Por último, y a modo de ejemplificación, se procederá a realizar una **simulación**, por medio de Matlab, de un ejemplo práctico en el que se intenta filtrar una señal recibida por GPS en el caso de un movimiento rectilíneo. De esta forma, se intentarán ilustrar las dificultades que existen a la hora del modelado de un sistema, puesto que se deben tener en cuenta una gran cantidad de parámetros y traspasar un problema real a un modelo matemático no es tarea sencilla, y por ello, se cometerán errores que también serán necesarios incluir en el modelo. Este ejemplo tiene como objetivo ilustrar el funcionamiento del filtro de Kalman, puesto que un análisis teórico de dicha herramienta no se considera suficiente para entenderlo lo mejor posible. De esta forma, se puede observar en un ejemplo real la forma que tienen las ecuaciones que componen el algoritmo del filtro, así como las diferentes matrices que las componen.

2 | Modelado en el Espacio de Estados

El método de espacio de estados está basado en la descripción del sistema mediante n ecuaciones en diferencias, que se agrupan en una ecuación vectorial matricial en diferencias [13].

El **Estado** de un sistema dinámico es el conjunto más pequeño de variables que permiten predecir la evolución del sistema conocidas las entradas, es decir, el conocimiento de esas “variables de estado” en un determinado instante t_0 junto con el conocimiento de los valores de la señal de entrada para los instantes $t > t_0$, permite determinar el comportamiento y evolución del sistema para cualquier instante de tiempo $t > t_0$. El espacio n -dimensional de todos los posibles valores del estado es el **Espacio de Estados**. Las variables de estado se agrupan en el *vector de estado*

La complejidad de los sistemas con los que usualmente se trabaja en ingeniería y en numerosos ámbitos aumenta enormemente. Los **métodos convencionales** más utilizados son, como ejemplos, el *lugar geométrico de las raíces* o los de *respuesta en frecuencia*. Ambos son útiles cuando se trata de sistemas con una única entrada y una única salida. Puesto que estos métodos son conceptualmente bastante sencillos, no requieren una gran capacidad de cálculo para llevarlos a cabo. El principal inconveniente que tienen es que solamente son aplicables a sistemas lineales e invariantes en el tiempo con una entrada y una salida. Además, estos métodos convencionales no son aplicables al diseño de sistemas de control óptimo y adaptable, y no suelen contar con la opción de incluir condiciones iniciales en el diseño de un modelo.

Puesto que la dinámica del movimiento de un vehículo es altamente no lineal, posee muchas entradas y salidas interrelacionadas de manera compleja y es variante en el tiempo, es necesario introducir el método de **Espacio Estados**, puesto que éste no posee dichas limitaciones, como se verá a continuación. Algunas de las ventajas de esta representación incluye:

1. La notación es muy compacta. Incluso sistemas realmente complejos pueden ser representados con dos “simples” ecuaciones que se explicarán a continuación

$$\begin{aligned}x(k+1) &= Gx(k) + Hu(k) \\ y(k) &= Cx(k) + Du(k)\end{aligned}\tag{2.1}$$

El término “simple” se encuentra entre comillas porque, como se verá en los siguientes apartados, estas dos ecuaciones pueden contener una gran cantidad de componentes puesto que trabaja con matrices y vectores.

2. Puesto que todos los sistemas están representados con la misma notación, es relativamente fácil desarrollar técnicas generales que resuelvan estos sistemas.
3. Los ordenadores fácilmente simulan las ecuaciones de primer orden, y por tanto, serán muy útil a la hora de resolver este tipo de problemas.
4. Se puede utilizar para representar sistemas no lineales (aparte de los lineales), y también para aquellos en los que existen múltiples entradas y salidas, llamados **MIMO** (del inglés, *Multi-Input, Multi-Output*), aparte de los llamados **SISO** (del inglés, *Single-Input, Single-Output*). El sistema puede ser variante o invariante en el tiempo.
5. Ofrece una muy buena idea sobre el estado interno del sistema, ya que proporciona un modelo de análisis con un diseño general donde se expresan relaciones entre las descripciones internas y externas.
6. Las condiciones iniciales del sistema pueden ser diferentes de cero.

2.1. Representación de sistemas en el espacio estados

Con la representación en espacio de estados se obtiene la capacidad de conocer y controlar en cierta medida la dinámica interna de un sistema y su respuesta. Las variables contenidas en el vector de estado deben ser capaces en su conjunto de determinar las condiciones de la dinámica del sistema. Cabe destacar en este punto que es posible que existan varias representaciones en variables de estado para un mismo sistema.

Para una mejor comprensión de los conceptos que aquí se tratarán, se procede a definir brevemente los siguientes términos [4], [13]:

- **Sistema**, se entenderá como una relación entre entradas y salidas.
- **Sistema Determinista**, si a cada entrada le corresponde una y solo una salida
- **Sistema Monovariante**. Es aquel que solo tiene una entrada y una salida. Si el sistema tiene más de una entrada o más de una salida se le considerará multivariante.
- **Sistema Causal o No Anticipatorio**. Es aquel que su salida para cierto tiempo t_1 , no depende de entradas aplicadas después de t_1 . Obsérvese que la definición implica que un sistema no causal es capaz de predecir entradas futuras, por lo tanto la causalidad es una propiedad intrínseca de cualquier sistema físico.
- **Sistema Dinámico**. Es aquel cuya salida presente depende de entradas pasadas y presentes. Si el valor de la salida en t_1 depende solamente de la entrada aplicada en t_1 , el sistema se conoce como **estático** o sin memoria. La salida de un sistema estático permanece constante si la entrada no cambia. En un sistema dinámico la salida cambia con el tiempo aunque no se cambie la entrada, a menos que el sistema ya se encuentre en estado estable.
- **Sistema Invariante en el Tiempo**. Es aquel que tiene parámetros fijos o estacionarios con respecto al tiempo, es decir, sus características no cambian al pasar el tiempo o dicho de otra forma, sus propiedades son invariantes con traslaciones de tiempo.
- **Sistema No Lineal**. Un sistema es no lineal si no se aplica el principio de superposición. Por ello, para un sistema que sea no lineal, la respuesta a la suma de dos entradas no puede calcularse tratando por separado a cada entrada y sumando los resultados obtenidos.

La dinámica de un sistema se puede describir en función del valor del vector de estados y de la señal de entrada (asumiendo que el sistema es *no autónomo*) mediante las siguientes expresiones:

$$\begin{aligned}x(k+1) &= f(x(k), u(k), k) \\ y(k) &= g(x(k), u(k), k)\end{aligned}\tag{2.2}$$

donde la notación ξ_k indica el valor tomado por ξ en el instante de tiempo t_k y f y g pueden ser funciones de cualquier tipo. Otra posible representación de este tipo de sistemas es la siguiente:

$$\begin{aligned}x(k+1) &= Gx(k) + Hu(k) \\ y(k) &= Cx(k) + Du(k)\end{aligned}\tag{2.3}$$

la cual es utilizada para trabajar con sistemas lineales e invariantes en el tiempo (LTI), y donde

- $\mathbf{x}(k)$ = Vector de estado (dimensión n)
- $\mathbf{y}(k)$ = Vector de salida (dimensión m)

- $\mathbf{u}(k)$ = Vector de entrada (dimensión r)
- $\mathbf{G}(k)$ = Matriz de estado (dimensión $n \times n$)
- $\mathbf{H}(k)$ = Matriz de entrada (dimensión $n \times r$)
- $\mathbf{C}(k)$ = Matriz de salida (dimensión $m \times n$)
- $\mathbf{D}(k)$ = Matriz de transmisión directa (dimensión $m \times r$)

Las ecuaciones 2.3 también se pueden representar mediante el diagrama de bloques de la Figura 2.1

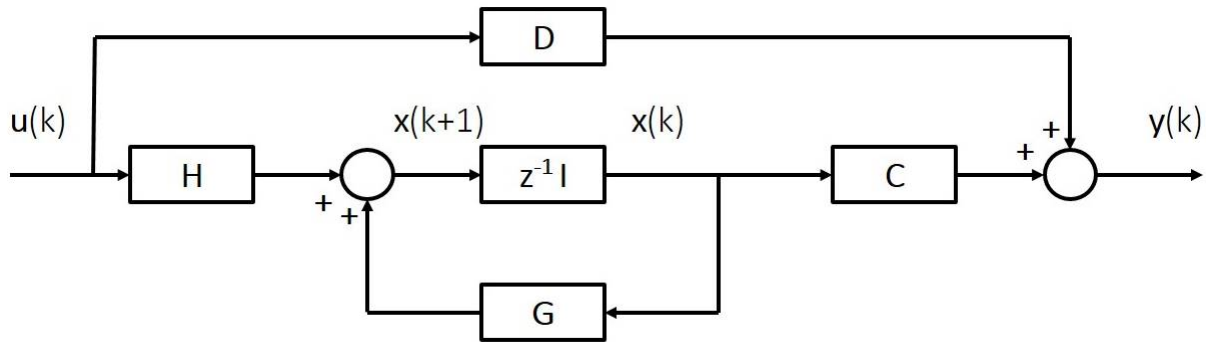


Figura 2.1: Diagrama de bloques de la representación en espacio estados de un sistema LTI [13]

2.2. Obtención de la representación en espacio de estados de sistemas discretos

Para comenzar esta sección, se explicará previamente la diferencia que existe entre un sistema continuo y un sistema discreto.

- **Sistema en tiempo Continuo.** Es aquel en el que las señales con las que trabaja el sistema son funciones de la variable continua tiempo t . Un ejemplo de señal continua $[x(t)]$ sería la variación de temperatura que se produce en los componentes de un sistema, la intensidad luminosa que se recibe del sol, señales sinusoidales, etc.
- **Sistema en tiempo Discreto.** Es aquel sistema en el cual una o más de las variables involucradas pueden cambiar solamente en valores concretos de tiempo. A dichos instantes se les denotará como k o kT , e indican los tiempos en los que se lleva a cabo alguna medición de tipo físico o el instante en el que se extraen datos de algún parámetro, variable u otro sistema. Es decir, una señal discreta $[x(kT)]$ solo toma valores para ciertos puntos, de forma que entre dos instantes en los que se toman valores se mantendrá el último valor tomado. Las señales digitales son el mejor ejemplo de señales discretas. Como ejemplos, las señales de GPS son discretas porque se envían cada cierto periodo de tiempo, los pulsos cardíacos y los semáforos, entre otras. En la sección 2.7 se explicará el proceso mediante el cual una señal continua podrá ser discretizada para su uso dentro del modelado en espacio estados. En el caso bajo estudio, de la simulación de la integración de dos señales de GPS y de IMU de un UAV, se utilizará un sistema en tiempo discreto. Esto será debido a que las señales tanto por parte del GPS como de la IMU se reciben en instantes de tiempo concretos, discretos.

Se puede decir que el sistema discreto bajo estudio es descrito por la siguiente expresión:

$$y(k) + a_1y(k-1) + a_2y(k-2) + \dots + a_ny(k-n) = b_0u(k) + b_1u(k-1) + \dots + b_nu(k-n) \quad (2.4)$$

donde $\mathbf{u}(\mathbf{k})$ es la entrada (con sus correspondientes coeficientes \mathbf{b}_j) e $\mathbf{y}(\mathbf{k})$ es la salida (con sus correspondientes coeficientes \mathbf{a}_j) del sistema en el instante de muestreo \mathbf{k} . Es posible que alguno de los coeficientes a_i ($i=1,2,\dots,n$) y b_j ($j=0,1,2,\dots,n$) pueden ser cero. En otras palabras, esta expresión significa que la salida del sistema actual $y(k)$ (lo que por ejemplo podría ser la posición actual del vehículo), depende tanto de las posiciones anteriores $y(k-n)$ y de las variables de entrada $u(k)$ (las cuales podrían ser una fuerza que desplazara el vehículo como el viento, la gravedad, una variación de velocidad desde los mandos de control, etc.).

A partir de su función de transferencia será posible obtener la representación de espacio de estados del sistema. Puesto que la función de transferencia proporciona la relación que existe entre las entradas y salidas del sistema, se obtiene:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_nz^{-n}}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n}} \quad (2.5)$$

Existen dos métodos para obtener la representación de espacio de estados a partir de 2.5: el **Método de programación directa** y el **Método de programación anidada**, los cuales se explicarán a continuación. [13] [16] [25].

2.2.1. Método de programación directa

Se reescribe la función de transferencia como:

$$G(z) = b_0 + \frac{(b_1 - a_1b_0)z^{-1} + (b_2 - a_2b_0)z^{-2} + \dots + (b_n - a_nb_0)z^{-n}}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n}} \quad (2.6)$$

y también teniendo en cuenta que $G(z) = \frac{Y(z)}{U(z)}$, se obtiene:

$$Y(z) = b_0U(z) + \frac{(b_1 - a_1b_0)z^{-1} + (b_2 - a_2b_0)z^{-2} + \dots + (b_n - a_nb_0)z^{-n}}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n}}U(z) \quad (2.7)$$

que puede reescribirse como:

$$Y(z) = b_0U(z) + \tilde{Y}(z)U(z) \Leftrightarrow \tilde{Y}(z) = \frac{(b_1 - a_1b_0)z^{-1} + (b_2 - a_2b_0)z^{-2} + \dots + (b_n - a_nb_0)z^{-n}}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n}} \quad (2.8)$$

Conociendo la expresión de $\tilde{Y}(z)$ se puede conseguir una variable auxiliar intermedia entre las entradas y salidas, $Q(z)$, a partir de la cual se podrán definir las relaciones entre el estado en un instante y en el instante anterior (como se verá en la formulación del modelo de espacio estados de la expresión 2.13), y que además cumple lo siguiente

$$Q(z) = \frac{\tilde{Y}(z)}{(b_1 - a_1b_0)z^{-1} + (b_2 - a_2b_0)z^{-2} + \dots + (b_n - a_nb_0)z^{-n}} = \frac{U(z)}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n}} \quad (2.9)$$

De ahí se obtiene que:

$$Q(z) = -a_1z^{-1}Q(z) - a_2z^{-2}Q(z) - \dots - a_nz^{-n}Q(z) + U(z) \quad (2.10)$$

$$\tilde{Y}(z) = (b_1 - a_1b_0)z^{-1}Q(z) + (b_2 - a_2b_0)z^{-2}Q(z) + \dots + (b_n - a_nb_0)z^{-n}Q(z) \quad (2.11)$$

Las variables de estado del problema serían:

$$\begin{aligned}
 X_1(z) &= z^{-n}Q(z) \\
 X_2(z) &= z^{-(n-1)}Q(z) \\
 &\dots \\
 X_n(z) &= z^{-1}Q(z)
 \end{aligned}$$

Utilizando las propiedades de la transformada de Z:

$$\begin{aligned}
 zX_1(z) = X_2(z) &\Leftrightarrow x_1(k+1) = x_2(k) \\
 zX_2(z) = X_3(z) &\Leftrightarrow x_2(k+1) = x_3(k) \\
 &\dots \Leftrightarrow \dots \\
 zX_{(n-1)}(z) = X_n(z) &\Leftrightarrow x_{(n-1)}(k+1) = x_n(k)
 \end{aligned}$$

Teniendo en cuenta las expresiones anteriores, es posible obtener $Q(z) = zX_n(z)$, y por tanto se puede reescribir la expresión de 2.10 como:

$$\begin{aligned}
 zX_n(z) &= -a_1X_n(z) - a_2X_{(n-1)}(z) - \dots - a_nX_1(z) + U(z) \Leftrightarrow \\
 x_n(k+1) &= -a_nx_1(k) - a_{n-1}x_2(k) - \dots - a_1x_n(k) + u(k)
 \end{aligned} \tag{2.12}$$

Con esta información ya es posible obtener aquello que se estaba buscando: la expresión de la ecuación de estado. De forma matricial, quedaría de la siguiente forma:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{(n-1)}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{(n-1)}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u_k \tag{2.13}$$

En la ecuación 2.13 se puede observar cómo el siguiente estado ($x_1(k+1)$) depende del anterior ($x_2(k)$). Esta ecuación es una expresión bastante clara donde se pueden observar los estados internos del sistema, de una forma bastante visual, fenómeno que no se producía cuando se tenía cuando se observaba directamente la función de transferencia. La ecuación 2.13 es la nombrada anteriormente como “*Ecuación de estado*”. Si del sistema bajo estudio se quieren analizar los datos de la posición y velocidad (estados), por ejemplo, pues esta expresión sería la necesaria para calcular dichos datos, los cuales se basan en los estados anteriores y en las entradas.

Para calcular ahora la denominada “*Ecuación de salida*” procedente de la función de transferencia anterior, simplemente bastaría con reescribir la ecuación 2.11:

$$\begin{aligned}
 \tilde{Y}(z) &= (b_1 - a_1b_0)X_n(z) + (b_2 - a_2b_0)X_{(n-1)}(z) + \dots + (b_n - a_nb_0)X_1(z) \\
 &\Downarrow \textit{antitransformando} \\
 y(k) &= (b_n - a_nb_0)x_1(k) + (b_{n-1} - a_{n-1}b_0)x_2(k) + \dots + (b_1 - a_1b_0)x_n(k) + b_0u(k)
 \end{aligned} \tag{2.14}$$

Se llega a una expresión de la ecuación de la salida de la siguiente forma:

$$y(k) = \begin{bmatrix} b_n - a_n b_0 & b_{n-1} - a_{n-1} b_0 & \cdots & b_1 - a_1 b_0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{(n-1)}(k) \\ x_n(k) \end{bmatrix} + b_0 u(k) \quad (2.15)$$

Como se ha dicho anteriormente al principio de esta sección, existen varias representaciones de espacio de estados para un mismo sistema. Las ecuaciones de estado 2.13 y de salida 2.15 son un ejemplo de una de esas representaciones. Ambas representan el espacio estados del sistema cuya función de transferencia es la expresión 2.5, que se denomina *forma canónica controlable (FCC)*. Más adelante se explicarán con más detalle este tipo de representación.

2.2.2. Método de programación anidada

Se parte en este caso de la misma función de transferencia 2.5, para encontrar otro tipo de representación del espacio de estados.

$$\begin{aligned} Y(z) - b_0 U(z) + z^{-1}(a_1 Y(z) - b_1 U(z)) + \dots + z^{-n}(a_n Y(z) - b_n U(z)) &= 0 \Leftrightarrow \\ Y(z) &= b_0 U(z) + z^{-1}(b_1 U(z) - a_1 Y(z) + z^{-1}(b_2 U(z) - a_2 Y(z) + z^{-1}(b_3 U(z) - a_3 Y(z) + \dots))) \Leftrightarrow \\ Y(z) &= b_0 U(z) + X_n(z) \implies \text{antitransformando} \implies y(k) = x_n(k) + b_0 u(k) \end{aligned} \quad (2.16)$$

A partir de esta expresión se pueden definir las siguientes variables de estado:

$$\begin{aligned} X_n(z) &= z^{-1}(b_1 U(z) - a_1 Y(z) + X_{(n-1)}(z)) \\ X_{(n-1)}(z) &= z^{-1}(b_2 U(z) - a_2 Y(z) + X_{(n-2)}(z)) \\ &\dots \\ X_2(z) &= z^{-1}(b_{n-1} U(z) - a_{n-1} Y(z) + X_1(z)) \\ X_1(z) &= z^{-1}(b_n U(z) - a_n Y(z)) \end{aligned} \quad (2.17)$$

Si se sustituye la última expresión de 2.16 en las variables de estado, y a su vez se multiplica por z en ambos miembros de la igualdad:

$$\begin{aligned} zX_n(z) &= X_{(n-1)}(z) - a_1 X_n(z) + (b_1 - a_1 b_0)U(z) \\ zX_{(n-1)}(z) &= X_{(n-2)}(z) - a_2 X_n(z) + (b_2 - a_2 b_0)U(z) \\ &\dots \\ zX_2(z) &= X_1(z) - a_{n-1} X_n(z) + (b_{n-1} - a_{n-1} b_0)U(z) \\ zX_1(z) &= -a_n X_n(z) + (b_n - a_n b_0)U(z) \end{aligned} \quad (2.18)$$

Antitransformando la ecuación 2.18:

$$\begin{aligned} x_1(k+1) &= -a_n x_n(k) + (b_n - a_n b_0)u(k) \\ x_2(k+1) &= x_1(k) - a_{n-1} x_n(k) + (b_{n-1} - a_{n-1} b_0)u(k) \\ &\dots \\ x_{(n-1)}(k+1) &= x_{(n-2)}(k) - a_2 x_n(k) + (b_2 - a_2 b_0)u(k) \\ x_n(k+1) &= x_{(n-1)}(k) - a_1 x_n(k) + (b_1 - a_1 b_0)u(k) \end{aligned} \quad (2.19)$$

Teniendo en cuenta la antitransformada que se obtuvo en la ecuación 2.16 y 2.19, se pueden obtener la representación de espacio de estados de el sistema bajo estudio.

$$\begin{aligned}
 \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{(n-1)}(k+1) \\ x_n(k+1) \end{bmatrix} &= \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & -a_n \\ 1 & 0 & \cdots & 0 & 0 & -a_{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & -a_2 \\ 0 & 0 & \cdots & 0 & 1 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{(n-1)}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} b_n - a_n b_0 \\ b_{n-1} - a_{n-1} b_0 \\ \vdots \\ b_2 - a_2 b_0 \\ b_1 - a_1 b_0 \end{bmatrix} u(k) \\
 y(k) &= [0 \quad 0 \quad \cdots \quad 0 \quad 1] \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{(n-1)}(k) \\ x_n(k) \end{bmatrix} + b_0 u(k)
 \end{aligned} \tag{2.20}$$

A este tipo de representación, donde en la expresión 2.20 se aprecian la ecuación de estado y de salida, se le denomina *forma canónica observable (FCO)*. Se estudiará más adelante por qué motivos se utilizará esta forma de representación o la anterior. Cuando se realiza un análisis y diseño de un modelo concreto en el dominio de estado, típicamente se transforman las ecuaciones de las que se dispone en alguna de estas formas particulares debido a las ventajas que ofrecen. Por ejemplo, la FCC posee propiedades interesantes que la hacen conveniente para **pruebas de controlabilidad**, mientras que la FCO se utiliza más para **pruebas de observabilidad**. Ambos conceptos (controlabilidad y observabilidad) serán introducidos en la sección 2.8.

2.3. Relación entre la representación en espacio de estados y la función de transferencia

El modelado y control de sistemas basado en la transformada de Laplace ofrece un enfoque sencillo y de fácil aplicación. Permite analizar sistemas utilizando una serie de reglas algebraicas en lugar de trabajar con ecuaciones diferenciales. Pero no en todas sus formas tiene la misma elegancia. Las **funciones de transferencia** cuentan con una serie de **limitaciones** a la hora de describir un sistema [4]:

- No proporciona información sobre la estructura física del sistema
- Sólo es válida para sistemas lineales con una entrada y una salida e invariantes en el tiempo. Este es uno de los motivos por los cuales se descarta trabajar con funciones de transferencia y se opta por el espacio estados. En el sistema bajo estudio, en el cual se integran señales de sistemas GPS e IMU, las señales recibidas son discretas (con las que trabaja el modelado en espacio estados) y no continuas (con las que trabaja la transformada de Laplace). Como se verá en el apartado de Sensores, existen múltiples entradas y múltiples salidas (se obtienen señales de altímetros, velocímetros, cámaras, codificadores, giróscopos, acelerómetros, magnetómetros, etc.) que se querrán integrar unas con otras para obtener el mejor resultado posible, lo cual no puede llevarse a cabo con una función de transferencia (trabaja con una única entrada y única salida).
- No proporciona información de lo que pasa dentro del sistema. Si se quiere obtener un dato intermedio de alguno de los sensores o sistemas que se encuentran operativos en el proceso, es más complejo obtenerlo en el caso de trabajar con este tipo de descripciones.
- Se necesita que las condiciones iniciales del sistemas sean nulas. De hecho, la función de transferencia de un sistema lineal invariante en el tiempo se define como *la transformada de Laplace de*

la respuesta al impulso, con todas las condiciones iniciales iguales a cero. En el caso del vehículo bajo estudio, por ejemplificar, se quieren tomar datos de una trayectoria rectilínea con velocidad constante de 8m/s. Es decir, comenzando y terminando con esa velocidad. En este caso la velocidad inicial no es nula, y por tanto, no se podría modelar este proceso mediante la función de transferencia. Otro motivo por el cual se utilizará el modelado en espacio estados, porque tal y como se ha demostrado, no limita la maniobrabilidad de actuaciones que se pueden llevar a cabo

No solamente el caso bajo estudio se encuentra fuera del rango de actuación de la función de transferencia, sino que la mayoría de sistemas dinámicos no cumplen con dichos requisitos. Los **sistemas reales**, por lo general, presentan no linealidades, cuentan con más de una entrada y salida, sus parámetros cambian con el tiempo y sus condiciones iniciales no siempre son cero.

Con un sencillo análisis de la representación de espacio de estados se puede obtener la función de transferencia.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \implies \text{Laplace} \implies \begin{aligned} sX(s) &= AX(s) + BU(s) \\ Y(s) &= CX(s) + DU(s) \end{aligned} \quad (2.21)$$

De la ecuación de estado se puede obtener:

$$(sI - A)X(s) = BU(s) \implies X(s) = (sI - A)^{-1}BU(s) \quad (2.22)$$

Y sustituyéndolo en la ecuación de salida:

$$Y(s) = [C(sI - A)^{-1}B + D]U(s) \implies G(s) = \frac{Y(s)}{U(s)} = [C(sI - A)^{-1}B + D] \quad (2.23)$$

Pero lo interesante en este tema no es obtener la función de transferencia a partir de la representación de espacio estados, sino al contrario. Como se ha repetido anteriormente, el proceso de convertir de función de transferencia a espacio de estados no es único. Se puede decir que todas las transformaciones son "equivalentes", puesto que las propiedades del sistema no cambian. Lo atractivo de este proceso es que algunas representaciones de espacio de estados pueden tener mas ventajas que otras dependiendo del caso para una tarea particular. Algunas posibles representaciones son [14]:

1. Forma canónica controlable (*First companion form*)
2. Forma canónica observable (*Second companion form*)
3. Forma canónica de Jordan
4. Forma canónica controlable alternativa / Forma canónica Diagonal (*Alternative first companion form, Toeplitz first companion form*)

En contraposición a los inconvenientes de la función de transferencia, el espacio de estados presenta una serie de ventajas [4], como se mencionó anteriormente:

- Aplicable a sistemas lineales y no lineales
- Permite analizar sistemas de más de una entrada y una salida
- Pueden ser variantes o invariantes en el tiempo
- Las condiciones iniciales no necesariamente deben ser nulas

- Es capaz de proporcionar información de lo que está sucediendo en el interior del sistema en cada momento
- Los resultados que ofrece los presenta de una forma sencilla y elegante, lo cual es un punto a favor en cuanto a una mejor comprensión del estudio. Este punto parece de poca importancia, pero cuando se tratan sistemas complejos en los que se tienen múltiples datos interrelacionados entre sí, una buena visualización ayuda bastante a su mejor entendimiento.

2.4. No unicidad de la representación en espacio de estados de un sistema

Como ya se viene advirtiendo en los apartados anteriores y se ha comprobado con los dos métodos analizados (*Método de programación directa* y *Método de programación anidada*), a un sistema descrito por su función de transferencia le corresponden al menos dos representaciones en espacio de estados diferentes (*FCC* y *FCO* respectivamente). Esto es debido principalmente al hecho de que la dinámica de un sistema puede ser descrita por multitud de variables de estado. Se pueden tomar variables de estado que sean combinaciones lineales de otras, que no alteren las propiedades del sistema, pero que sin embargo obliguen a modificar la estructura de resolución del problema. En este caso se obtendrían unas representaciones distintas pero equivalentes.

Para demostrar este fenómeno, se utilizará una transformación mediante una matriz invertible P , la cual relacionará el actual vector de estado $x(k)$ con otro $\tilde{x}(k)$ con variables de estado distintas mediante [13]:

$$x(k) = P\tilde{x}(k) \quad (2.24)$$

Se obtendría una nueva ecuación de estado del sistema:

$$P\tilde{x}(k+1) = GP\tilde{x}(k) + Hu(k) \implies \tilde{x}(k+1) = P^{-1}GP\tilde{x}(k) + P^{-1}Hu(k) \quad (2.25)$$

Por lo que las ecuaciones que definen la nueva representación del sistema serían las siguientes:

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{G}\tilde{x}(k) + \tilde{H}u(k) & \tilde{G} &= P^{-1}GP, & \tilde{H} &= P^{-1}H \\ y(k) &= \tilde{C}\tilde{x}(k) + \tilde{D}u(k) & \tilde{C} &= CP, & \tilde{D} &= D \end{aligned} \quad (2.26)$$

Se obtiene así un sistema en espacio de estados equivalente al anterior, pero con variables diferentes.

2.5. Resolución de las ecuaciones del espacio de estados

Para la resolución de estas ecuaciones, es importante primero saber cuáles son los requisitos para su obtención. Éstas se pueden obtener mediante las ecuaciones diferenciales que representan un sistema. Los pasos a seguir a grandes rasgos son los siguientes [4] [17]:

1. Identificar las leyes o teorías que gobiernan el comportamiento que sigue el sistema (Leyes de termodinámica, Leyes dinámicas, **Segunda Ley de Newton**, Ley de voltajes y corrientes de Kirchoff, Ley de Ampere, Ley de Ohm, Ley de Boyle, etc.)
2. Seleccionar las variables de estado. Son las mínimas variables que determinan el comportamiento dinámico del sistema. En este paso se determina el orden n del sistema que se va a estudiar.

3. Encontrar la dinámica de cada estado. Es decir, se debe conocer cómo varía esa variable con respecto al tiempo, o lo que es lo mismo, su derivada con respecto al tiempo. En este paso son definidas las matrices A, B, C y D del sistema representado en 2.21.

En este trabajo se explicarán tres métodos para obtener el valor del vector de estado. A partir del valor inicial x_0 , se obtendrá el valor para cualquier instante de tiempo $k > 0$, mediante alguno de los siguientes procesos:

- Procedimiento recursivo
- Matriz de transición de estados
- Método basado en la transformada Z

2.5.1. Procedimiento recursivo

Este procedimiento debe su nombre al hecho de que es necesario realizar un proceso iterativo para obtener la solución. Si se realiza dicho proceso sobre las ecuaciones 2.3, las cuales representan un sistema LTI a partir de $k = 0$:

$$\begin{aligned}x(1) &= Gx(0) + Hu(0) \\x(2) &= Gx(1) + Hu(1) = G^2x(0) + GHu(0) + Hu(1) \\x(3) &= Gx(2) + Hu(2) = G^3x(0) + G^2Hu(0) + GHu(1) + Hu(2) \\&\dots\end{aligned}\tag{2.27}$$

Lo cual generalizado para cualquier $k > 0$ quedaría de la siguiente forma:

$$x(k) = G^k x(0) + \sum_{j=0}^{k-1} G^{k-j-1} Hu(j)\tag{2.28}$$

Simplemente observando la ecuación 2.28 se puede observar que los valores del actual $x(k)$ dependerán tanto del estado inicial x_0 como de los valores de la entrada $u(j)$. La expresión 2.29 determina la salida del sistema.

$$y(k) = CG^k x(0) + C \sum_{j=0}^{k-1} G^{k-j-1} Hu(j) + Du(k)\tag{2.29}$$

2.5.2. Matriz de transición de estados

Este proceso es un poco más limitado que el anterior, puesto que se presupone que no existe una señal de entrada $u(k)$, y por tanto el estado actual solamente dependería del anterior. Convirtiendo esta información en una ecuación matemática se obtiene lo siguiente:

$$x(k+1) = Gx(k)\tag{2.30}$$

Puesto que no posee una señal de entrada, se puede expresar la solución de la ecuación refiriéndose al estado inicial, puesto que éste es el único que se debe conocer para ir averiguando los siguientes estados actuales. Es decir, se necesita una función que, al conocer el momento en el que queremos calcular el estado, sea capaz de calcular dicho estado a partir del instante inicial. A esta función se le llamará ψ :

$$x(k) = \psi_k x(0) \implies \text{con : } \psi(k+1) = G\psi(k) \text{ y } \psi(0) = I \implies \text{es decir : } \psi(k) = G^k \quad (2.31)$$

A dicha función ψ_k se le denomina *matriz de transición de estados*, y contiene toda la información sobre movimientos libres del sistema descrito por 2.30. Estos movimientos libres son de los que se hablaba anteriormente, y se refieren a los cambios de estado o su evolución en ausencia de la entrada u_k .

Las soluciones de la ecuación de estado y de la ecuación de salida son las siguientes:

$$\begin{aligned} x(k) &= \psi(k)x(0) + \sum_{j=0}^{k-1} \psi(k-j-1)Hu(j) \\ &= \psi(k)x(0) + \sum_{j=0}^{k-1} \psi(j)Hu(k-j-1) \\ y(k) &= C\psi(k)x(0) + C \sum_{j=0}^{k-1} \psi(j)Hu(k-j-1) + Du(k) \end{aligned} \quad (2.32)$$

2.5.3. Método basado en la transformada Z

Este método es complejo debido a que se necesita calcular la transformada de algunas de las expresiones. Como se ha dicho con anterioridad, la transformada Z difiere de la transformada de Laplace en que la primera se utiliza para sistemas que hacen uso de señales discretas (modelo espacio estados), mientras que el segundo se utiliza con señales continuas (función de transferencia, por ejemplo). Partiendo de 2.3 y realizando la transformada Z a ambos lados de la igualdad se obtiene:

$$\begin{aligned} zX(z) - zx(0) &= GX(z) + HU(z) \Leftrightarrow (zI - G)X(z) = zx(0) + HU(z) \\ \Leftrightarrow X(z) &= (zI - G)^{-1}zx(0) + (zI - G)^{-1}HU(z) \end{aligned} \quad (2.33)$$

cuya antitransformada sería la siguiente:

$$x(k) = Z^{-1}[(zI - G)^{-1}z]x(0) + Z^{-1}[(zI - G)^{-1}HU(z)] \quad (2.34)$$

Si se compara esta expresión con la obtenida en el método recursivo 2.28, se pueden igualar algunos términos, con lo que quedaría:

$$G^k = Z^{-1}[(zI - G)^{-1}z] \quad \text{y} \quad \sum_{j=0}^{k-1} G^{k-j-1}Hu(j) = Z^{-1}[(zI - G)^{-1}HU(z)] \quad (2.35)$$

2.6. Linealización de las ecuaciones de estado

La mayoría de los procesos que suceden en la naturaleza contienen un alto grado de no linealidad. Por este motivo, es importante que la ciencia y la técnica sean capaces de proporcionar métodos de resolución para este tipo de problemas. Puesto que la no linealidad de un fenómeno concreto tiene unas características muy particulares, desarrollar técnicas capaces de resolver un determinado tipo de

problema conllevaría horas de trabajo y esfuerzo para cada uno de ellos. Esto quiere decir que no existe una técnica o modelado general capaz de resolver todos los problemas no lineales. Es por ello que se desarrollan técnicas de linealización para poder abordar el mayor número posible de situaciones. Es importante destacar que solamente en circunstancias muy particulares se es capaz de obtener una solución exacta de unas ecuaciones diferenciales no lineales, y por ello la mayoría de los métodos de resolución optan por utilizar aproximaciones.

La linealización es un proceso mediante el cual un conjunto de ecuaciones diferenciales no lineales se *aproximan* por un conjunto lineal. Esta aproximación se puede entender de la siguiente forma, parafraseando la observación de Baruch: “*Si la única herramienta disponible es un martillo, haz que todos los problemas parezcan clavos*”. [8]

Incluso se ha estudiado que los llamados “sistemas lineales” solamente lo son en rangos de operación limitados. Muchos de los sistemas electromecánicos, hidráulicos, neumáticos, etc., conllevan relaciones no lineales entre las variables. Por ejemplo, los amortiguadores en algunos sistemas pueden ser lineales para operaciones a baja velocidad, pero fácilmente pueden volverse no lineales a velocidades más altas, y la fuerza de amortiguamiento puede hacerse proporcional al cuadrado de la velocidad de operación. [13]

Una posible representación de un sistema no lineal mediante un conjunto de ecuaciones similares a los usados anteriormente podría ser:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), & x(t_0) &= x(0) \\ y(t) &= h(x(t)) \end{aligned} \tag{2.36}$$

donde se ha utilizado la notación en derivadas temporales para una mejor comprensión de ecuación diferencial. La solución de $x(t)$ de la ecuación 2.36 es una curva en el espacio. Se conoce como *trayectoria de estado*, donde la variable t varía desde cero hasta infinito.

A la hora de linealizar un sistema, se debe tener en cuenta la desviación producida sobre el punto de operación. El sistema lineal será equivalente al sistema no lineal dentro de un rango de operación limitado. Para el tamaño relativo de la desviación, se tienen en cuenta dos factores:

- La naturaleza de la no linealidad, esto es, lo lejos que se encuentra el sistema de ser lineal.
- El grado de exactitud que se desea obtener entre el modelo real y el linealizado. Normalmente, los requisitos sobre la exactitud para el modelo, dependen de la aplicación del mismo.

2.6.1. Interpretación Gráfica

La interpretación gráfica de la linealización es la siguiente: línea recta que pasa tangente en uno de sus puntos (punto de operación). Se obtiene a partir de las derivadas parciales de las ecuaciones que describen el sistema dinámico con respecto a sus variables. Dicha tangente describe la función en cuestión de manera lineal en el punto de operación, pero si se estudiara cualquier otro punto de la curva, alejado del primero, se debería realizar otra linealización, puesto que la anterior solamente es útil en las proximidades del punto respecto al cual se aproximó. En la Figura 2.2 se representa un ejemplo de linealización.

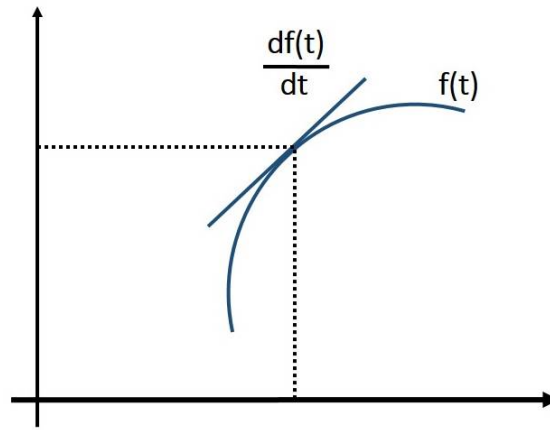


Figura 2.2: Ejemplo de linealización de una curva en un punto

2.6.2. Aproximación lineal del modelo

En este apartado se introducirá brevemente el desarrollo de la función no lineal en series de Taylor alrededor del punto de operación como método de linealización. Se considerará que los términos de orden superior son suficientemente pequeños (sus variaciones sólo se desvían ligeramente de la condición de operación), y por lo tanto no se considerarán en el desarrollo en serie de Taylor.

Se considera el sistema que se viene usando con anterioridad, donde $x(t)$ es la entrada e $y(t)$ es la salida. La relación entre ambas se obtiene mediante

$$y = f(x) \quad (2.37)$$

Considerando que la condición en la que se está trabajando corresponde con \bar{x} y \bar{y} , la ecuación 2.37 se expande en serie de Taylor alrededor de este punto:

$$\begin{aligned} y &= f(x) \\ &= f(\bar{x}) + \frac{df}{dx}(x - \bar{x}) + \frac{1}{2!} \frac{d^2f}{dx^2}(x - \bar{x})^2 + \dots \end{aligned} \quad (2.38)$$

donde las derivadas $\frac{df}{dx}$, $\frac{d^2f}{dx^2}$, ... se evalúan en $x = \bar{x}$. Si la variación $x = \bar{x}$ es pequeña, es posible no considerar los términos de orden superior en $x = \bar{x}$. Por lo tanto, la expresión 2.38 se escribiría tal que

$$\begin{aligned} y &= \bar{y} + K(x - \bar{x}), \implies \text{donde} \\ \bar{y} &= f(\bar{x}) \\ K &= \left. \frac{df}{dx} \right|_{x=\bar{x}} \end{aligned} \quad (2.39)$$

lo cual puede reescribirse como

$$y - \bar{y} = K(x - \bar{x}) \quad (2.40)$$

La primera conclusión que se puede obtener de analizar la expresión 2.40 es que $y - \bar{y}$ es proporcional a $x - \bar{x}$. Dicha ecuación también proporciona un **modelo matemático lineal** para un sistema **inicialmente no lineal**, en los alrededores del punto de operación $x = \bar{x}$, $y = \bar{y}$.

Para un sistema no lineal cuya salida y es una función de dos entradas x_1 y x_2

$$y = f(x_1, x_2) \quad (2.41)$$

se obtiene la siguiente expansión en series de Taylor alrededor del punto de operación \bar{x}_1, \bar{x}_2

$$y = f(\bar{x}_1, \bar{x}_2) + \left[\frac{\partial f}{\partial x_1}(x_1 - \bar{x}_1) + \frac{\partial f}{\partial x_2}(x_2 - \bar{x}_2) \right] + \frac{1}{2!} \left[\frac{\partial^2 f}{\partial x_1^2}(x_1 - \bar{x}_1)^2 + 2 \frac{\partial^2 f}{\partial x_1 \partial x_2}(x_1 - \bar{x}_1)(x_2 - \bar{x}_2) + \frac{\partial^2 f}{\partial x_2^2}(x_2 - \bar{x}_2)^2 \right] + \dots \quad (2.42)$$

Las derivadas parciales se evaluaría en $x_1 = \bar{x}_1, x_2 = \bar{x}_2$. Igual que anteriormente, los términos de orden superior pueden ser despreciados, con lo que el sistema matemático quedaría de la siguiente forma

$$y - \bar{y} = K_1(x_1 - \bar{x}_1) + K_2(x_2 - \bar{x}_2), \implies \text{donde}$$

$$\bar{y} = f(\bar{x}_1, \bar{x}_2)$$

$$K_1 = \left. \frac{\partial f}{\partial x_1} \right|_{x_1=\bar{x}_1, x_2=\bar{x}_2} \quad (2.43)$$

$$K_2 = \left. \frac{\partial f}{\partial x_2} \right|_{x_1=\bar{x}_1, x_2=\bar{x}_2}$$

2.7. Discretización de las ecuaciones de estado continuas

Es frecuente realizar las definiciones de los parámetros y el análisis de las ecuaciones en tiempo continuo, pero en muchos casos prácticos nos encontramos con dos situaciones típicas

1. Un sistema que contiene componentes en tiempo continuo, pero que debido a operaciones de muestreo y retención que realiza el diseñador, se convierten en señales discretas con respecto al tiempo. En este caso las ecuaciones que describen el sistema son ecuaciones diferenciales, las cuales deben ser convertidas en ecuaciones en diferencias o discretas.

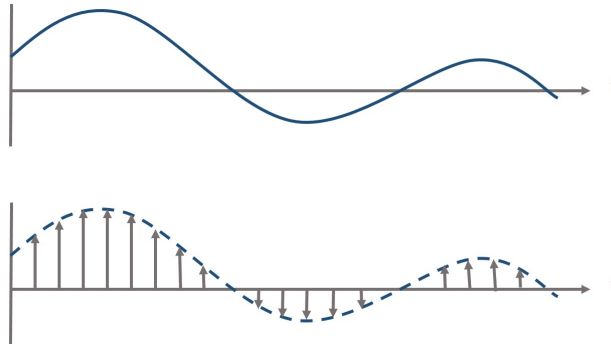


Figura 2.3: Muestreo de un sistema en tiempo continuo

El error introducido por la discretización se puede hacer despreciar usando una frecuencia de muestreo lo suficientemente alta (período de muestreo suficientemente bajo) en comparación con la constante de tiempo más significativa del sistema. Este hecho es demostrado gracias al **Teorema de Muestreo** (o **Teorema de Nyquist**), el cual afirma que es posible muestrear una señal de tiempo continuo gracias al uso de un tren de pulsos de frecuencia de muestreo $\omega_s = 2\pi/T$, con T como período de muestreo, siempre que dicha frecuencia de muestreo sea mayor que dos

veces el ancho de banda de la señal $\omega_s > 2\omega_M$. En el caso de no cumplirse esta condición, la señal no podrá ser muestreada debido a un fenómeno conocido como *aliasing* o solapamiento, tal y como se puede observar en la Figura 2.4

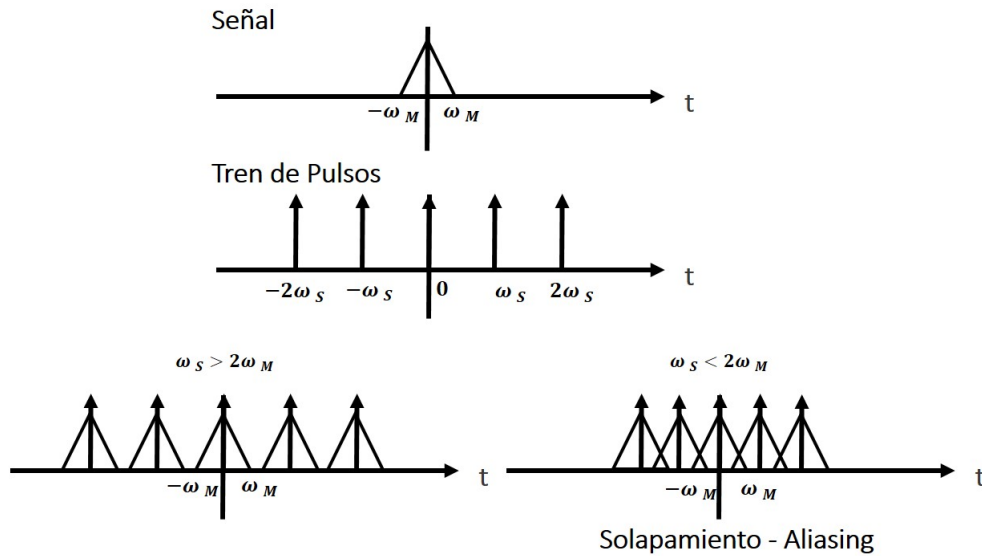


Figura 2.4: Ejemplificación del Teorema de Muestreo o Teorema de Nyquist

- Otro caso es aquel sistema completamente discreto con respecto al tiempo, y la dinámica debe contener ecuaciones en diferencias desde un principio.

En este apartado se presentará un método para convertir las ecuaciones en el espacio de estados continuas a discretas. Se considerará el sistema lineal e invariante en el tiempo (LTI) que se viene estudiando con anterioridad. Se presupondrá también que el vector de entrada $u(t)$ cambia sólo en instantes de muestreo uniformemente espaciados, es decir, solamente podrá cambiar en $t = kT$, para $k = 0, 1, 2, \dots$. El sistema a tener en cuenta es el siguiente

$$\begin{aligned} \dot{x} &= Ax + Bu \implies \text{discretizando} \implies x((k+1)T) = G(T)x(kT) + H(T)u(kT) \\ y &= Cx + Du \end{aligned} \quad (2.44)$$

donde puede observarse que las matrices G y H dependen del período de muestreo T . Una vez determinado dicho período, G y H son matrices constantes. Para determinar el valor de $G(T)$ y $H(T)$ se hará uso de la solución de la ecuación de estado en tiempo continuo

$$x(t) = e^{At}x(0) + e^{At} \int_0^t e^{-A\tau} Bu(\tau) d\tau \quad (2.45)$$

Se supone que la entrada $u(t)$ es muestreada y alimentada a un retenedor de orden cero, y así todas sus componentes son constantes en el intervalo entre dos instantes de muestreo consecutivos cualesquiera, es decir

$$u(t) = u(kT) \quad \text{para} \quad kT \leq t \leq kT + T \quad (2.46)$$

En vista de que

$$\begin{aligned} x((k+1)T) &= e^{A(k+1)T}x(0) + e^{A(k+1)T} \int_0^{(k+1)T} e^{-A\tau} Bu(\tau) d\tau \\ x(kT) &= e^{AkT}x(0) + e^{AkT} \int_0^{kT} e^{-A\tau} Bu(\tau) d\tau \end{aligned} \quad (2.47)$$

Al multiplicar la segunda ecuación de 2.47 por e^{At} y restándola de la primera de 2.47 se obtiene

$$x((k+1)T) = e^{AT}x(kT) + e^{A(k+1)T} \int_{kT}^{(k+1)T} e^{-A\tau} Bu(\tau) d\tau \quad (2.48)$$

Dada la ecuación 2.46, se puede sustituir $u(\tau) = u(kT) = \text{constante}$ en esta última ecuación. Se observa que $u(t)$ puede tomar un valor en $t = kT + T, [u(kT + T)]$, que puede ser distinto de $u(kT)$. Este valor en $u(\tau)$ con $\tau = kT + T$, que es el límite superior de integración, no afecta en absoluto al valor de la integral de esta última ecuación, ya que el integrando no incluye funciones impulso. Por lo tanto, se puede aplicar este concepto y operando se obtiene

$$\begin{aligned} x((k+1)T) &= e^{AT}x(kT) + e^{AT} \int_0^T e^{-At} Bu(kT) dt \\ &= e^{AT}x(kT) + \int_0^T e^{A\lambda} Bu(kT) d\lambda \end{aligned} \quad (2.49)$$

donde $\lambda = T - t$. Definiendo los siguientes parámetros

$$\begin{aligned} G(T) &= e^{AT} \\ H(T) &= \left(\int_0^T e^{A\lambda} d\lambda \right) B \end{aligned} \quad (2.50)$$

la ecuación 2.49 se convertiría en

$$x((k+1)T) = G(T)x(kT) + H(T)u(kT) \quad (2.51)$$

la cual coincide con la discretización de la ecuación 2.44. Por tanto, las ecuaciones de 2.50 proporcionan la discretización que se buscaba, aportando $G(T)$ y $H(T)$. Ambas dependen del período de muestreo como se dijo anteriormente. La ecuación de salida se convertirá por tanto en

$$y(kT) = Cx(kT) + Du(kT) \quad (2.52)$$

donde las matrices C y D son constantes y no dependen del tiempo de muestreo T . Si la matriz A es no singular, entonces $H(T)$ se puede simplificar

$$H(T) = \left(\int_0^T e^{A\lambda} d\lambda \right) B = A^{-1}(e^{AT} - I)B = (e^{AT} - I)A^{-1}B \quad (2.53)$$

A continuación se muestran una serie de observaciones con respecto a este proceso de discretización. [13]

- Al presuponer que el vector de entrada $u(t)$ es constante entre dos instantes de muestreo consecutivos cualesquiera, la representación en tiempo discreto se puede obtener integrando la ecuación de estado en tiempo continuo sobre un período de muestreo. La ecuación 2.51 se conoce como *equivalente con retenedor de orden cero* de la ecuación $\dot{x} = Ax + Bu$ (continua).

- Cabe destacar que en todo proceso de discretización de un sistema continuo, se producen errores debidos a las aproximaciones. En la ecuación 2.51 no existe ningún tipo de aproximación siempre que el vector de entrada $u(t)$ sea constante entre dos instantes de tiempo consecutivos cualesquiera, tal y como se supuso en la suposición.
- Para $T \ll 1$, $G(T)/G(0) = e^{A0} = I$. Por tanto $G(T)$ se acerca a la matriz identidad conforme el período de muestreo se hace muy pequeño, lo cual tiene sentido.

Para calcular e^{AT} de manera sencilla se puede hacer uso de la equivalencia

$$e^{At} = \mathcal{L}^{-1}[(sI - A)^{-1}] \quad (2.54)$$

donde \mathcal{L}^{-1} indica la transformada de Laplace inversa. De manera práctica el método constaría del cálculo de $(sI - A)^{-1}$ y aplicar después la transformada de Laplace inversa a cada elemento de la matriz.

2.8. Controlabilidad y Observabilidad

En la teoría y práctica del Control Moderno, los conceptos presentados por primera vez por Kalman [10] de controlabilidad y observabilidad juegan un papel realmente importante. Sobre sus condiciones se deduce la existencia o ausencia de una solución de un problema de diseño de Control Óptimo, colocación de polos, etc. Básicamente, se les puede catalogar como la diferencia existente entre la teoría de **Control Óptimo** y la teoría de **Control Clásica**. En esta última, el diseñador no conoce a priori si existe solución del problema o no, y las técnicas y modelados utilizados se basan en el principio de prueba y error. En la teoría de Control Óptimo, por otra parte, se cuenta con criterios para determinar desde el inicio del estudio si la solución del diseño existe para los parámetros del sistema y los objetivos del diseño. Se puede afirmar que la solución a un problema puede no existir si el sistema bajo consideración no es controlable. Es posible que un modelo matemático no posea la propiedad de controlabilidad y observabilidad aunque el sistema físico del cual se obtuvo sí lo fuera.

Se mostrará a continuación que la condición de **controlabilidad** de un sistema está íntimamente relacionada con la existencia o ausencia de soluciones de la realimentación de estado con el propósito de ubicar valores característicos del sistema en forma arbitraria, es decir, con la existencia de una secuencia de actuaciones para llevar el sistema a un estado arbitrario.

*“Se dice que un sistema es **completamente controlable** si cada variable de estado del proceso se puede llegar a controlar para llegar a un cierto objetivo en un tiempo finito, a través de algún control no restringido $u(t)$. Si cualquiera de las variables de estado es independiente de la señal de control, entonces resulta imposible controlar esa variable de estado y, por lo tanto, el sistema es no controlable.”. [10] [13]*

De forma intuitiva, es sencillo entender que si una de las variables de estado es independiente del control $u(t)$, no habría forma de dirigir esta variable de estado en particular al estado deseado en un tiempo finito por medio de un esfuerzo de control. Y si ese estado no es controlable, el sistema no es completamente controlable. Un sistema no controlable es aquel que posee un subsistema que se encuentra desconectado físicamente de la entrada.

En cualquier caso, interesa que el sistema bajo estudio sea completamente controlable. En el caso de los vehículos no tripulados, se pretende que todo sea controlable puesto que se necesitan unas velocidades de respuesta rápidas para algunas acciones críticas. Cuanto se tratan de dispositivos con un nivel de exigencia tecnológica tan alto, no es concebible dejar alguna de las variables de estado sin controlar. ¿Qué ocurriría si no se fuese capaz de controlar la velocidad del vehículo mediante un mando de control? ¿O su inclinación? La incapacidad de manejarlos haría disminuir o erradicar su uso en las distintas aplicaciones para las que son creados.

El concepto de **observabilidad** está relacionado con la condición de “observación” o estimación de las variables de estado a partir de las variables de salida, las cuales son generalmente medibles. Tiene que ver con la posibilidad de determinar el valor del vector de estados de un sistema a partir de observaciones de las salidas y las entradas de dicho sistema.

“Se dice que un sistema es **completamente observable** si cada variable de estado del sistema afecta a alguna de las salidas”. [10] [13]

Con el fin de aclarar estos términos, se ilustra el siguiente diagrama de bloques, y una explicación de la aplicación de la terminología sobre controlabilidad y observabilidad.

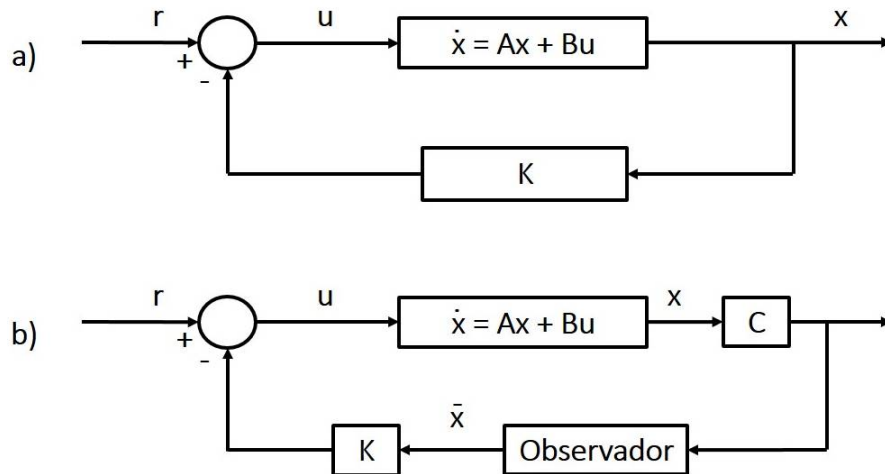


Figura 2.5: a) Sistema de Control con realimentación del estado. b) Sistema de control con realimentación del estado y observador.

La Figura 2.5 muestra un sistema con la dinámica descrita por la ecuación 2.55. Realimentando las variables de estado gracias a la matriz de ganancia \mathbf{K} , descrita en la ecuación 2.56, se obtiene un sistema en bucle cerrado que se describe mediante 2.57.

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.55)$$

$$u(t) = -Kx(t) + r(t) \quad (2.56)$$

$$\dot{x}(t) = (A - BK)x(t) + Br(t) \quad (2.57)$$

Este procedimiento forman las bases del diseño por ubicación de polos mediante el proceso de realimentación. El objetivo sería encontrar la matriz \mathbf{K} de realimentación, tal que los valores característicos de $(A - BK)$ tengan ciertos valores predeterminados a los que se pretende llegar. En este caso, se puede afirmar lo siguiente

1. El sistema de la ecuación 2.55 es **controlable** si existe una matriz de realimentación constante \mathbf{K} que permite que los valores característicos de $(A - BK)$ sean asignados de forma arbitraria. La controlabilidad juega un papel importante en la ubicación de polos en los sistemas de control.
2. Se puede dar el caso en el que no todas las variables de estado estén físicamente disponibles, y por tanto, se necesite implementar un “observador” que sea capaz de estimar el vector de estado a partir del vector de salida $y(t)$, tal y como se muestra en la Figura 2.5. El vector \bar{x} se denomina vector de estado observado, y se usa para generar el control $u(t)$ a través de la matriz de realimentación \mathbf{K} . La condición de que tal observador pueda ser diseñado para el sistema se conoce como **observabilidad** del sistema.

La controlabilidad se puede definir tanto para los estados (Controlabilidad del estado) como para la salida (Controlabilidad de la salida). Existen diferencias en su definición, y se explica a continuación para un sistema de control en tiempo discreto (ya que se ha explicado como discretizar sistemas continuos en la sección anterior), lineal e invariante en el tiempo.

2.8.1. Controlabilidad del estado completo

Se considera el sistema LTI de control definido por

$$x((k+1)T) = Gx(kT) + Hu(kT) \quad (2.58)$$

donde

- $\mathbf{x}(kT)$ = vector estado (dimensión n) en el k -ésimo instante de muestreo
- $\mathbf{u}(kT)$ = señal de control en el k -ésimo instante de muestreo. Se supone constante para $kT \leq t < (k+1)T$
- \mathbf{G} = matriz de $n \times n$
- \mathbf{H} = matriz de $n \times 1$
- T = período de muestreo

Se dice que un sistema es de **estado completamente controlable**, o simplemente de estado controlable, si existe una señal de control constante por intervalos $u(kT)$ definida a lo largo de un número finito de períodos de muestreo de forma que, partiendo de un estado inicial, el estado $x(kT)$ puede ser transferido al estado deseado x_f en n períodos de muestreo como máximo.

Se deduce a continuación la condición para la controlabilidad completa del estado

$$\begin{aligned} x(nT) &= G^n x(0) + \sum_{j=0}^{n-1} G^{n-j-1} H u(jT) \\ &= G^n x(0) + G^{n-1} H u(0) + G^{n-2} H u(T) + \dots + H u((n-1)T) \implies \\ x(nT) - G^n x(0) &= [\mathbf{H} : \mathbf{GH} : \dots : \mathbf{G}^{n-1} \mathbf{H}] \begin{bmatrix} \mathbf{u}((n-1)T) \\ \mathbf{u}((n-2)T) \\ \vdots \\ \mathbf{u}(0) \end{bmatrix} \end{aligned} \quad (2.59)$$

donde la matriz $M_c = [\mathbf{H} : \mathbf{GH} : \dots : \mathbf{G}^{n-1} \mathbf{H}]$ se denomina *matriz de controlabilidad*.

Puesto que H es una matriz $n \times 1$, se tiene que cada una de las matrices que componen la matriz de controlabilidad es una matriz de $n \times 1$ o un vector columna. La condición necesaria y suficiente para que el sistema sea completamente controlable es

$$\text{Rango}[\mathbf{H} : \mathbf{GH} : \dots : \mathbf{G}^{n-1} \mathbf{H}] = n \quad (2.60)$$

Se puede demostrar que esta condición también es válida para un sistema en el que $u(kT)$ sea un vector de dimensión r . [13]

2.8.2. Controlabilidad de la salida

Otro caso posible que se puede encontrar a la hora de diseñar un sistema de control, es preferir controlar la salida en vez del estado. Este tipo de controlabilidad no necesita que se produzca una controlabilidad completa del estado, y por esta razón, es necesaria definirla por separado. Es uno de los objetivos más comunes, controlar la evolución de la salida del sistema. En base a las ecuaciones

$$\begin{aligned}x((k+1)T) &= Gx(kT) + Hu(kT) \\ y(kT) &= Cx(kT) + Du(kT)\end{aligned}\tag{2.61}$$

donde

- $\mathbf{x}(kT)$ = vector estado (dimensión n) en el k -ésimo instante de muestreo.
- $\mathbf{u}(kT)$ = señal de control (dimensión r) en el k -ésimo instante de muestreo
- $\mathbf{y}(kT)$ = vector de salida (dimensión m) en el k -ésimo instante de muestreo
- \mathbf{G} = matriz de $n \times n$
- \mathbf{H} = matriz de $n \times r$
- \mathbf{C} = matriz de $m \times n$
- \mathbf{D} = matriz de $m \times r$

Se dice que el sistema 2.61 es de **salida completamente controlable** (o simplemente salida controlable) si es posible tener una señal de control no restringida $u(kT)$, definida en un conjunto finito de períodos de muestreo $0 \leq kT \leq nT$ tales que, partiendo de la salida inicial $y(0)$, la salida $y(kT)$ pueda ser transferida al punto deseado (punto arbitrario) y_f en el espacio de salidas, en n períodos de muestreo como máximo.

Se deduce a continuación las condiciones necesarias para que se produzca la controlabilidad de la salida. La salida puede ser dada por

$$\begin{aligned}y(nT) &= Cx(nT) + Du(nT) \\ &= CG^n x(0) + \sum_{j=0}^{n-1} CG^{n-j-1} Hu(jT) + Du(nT)\end{aligned}\tag{2.62}$$

obteniéndose

$$\begin{aligned}y(nT) - CG^n x(0) &= \sum_{j=0}^{n-1} CG^{n-j-1} Hu(jT) + Du(nT) \\ &= CG^{n-1} Hu(0) + CG^{n-2} Hu(T) + \dots + CHu((n-1)T) + Du(nT) \\ &= [\mathbf{D} : \mathbf{CH} : \mathbf{CGH} : \dots : \mathbf{CG}^{n-1} \mathbf{H}] \begin{bmatrix} \mathbf{u}(nT) \\ \mathbf{u}((n-1)T) \\ \dots \\ \mathbf{u}(0) \end{bmatrix}\end{aligned}\tag{2.63}$$

Una condición necesaria y suficiente para que el sistema sea de salida completamente controlable es que la matriz de la última expresión de 2.63 sea de rango m , es decir

$$\text{Rango}[\mathbf{D}:\mathbf{C}\mathbf{H}:\mathbf{C}\mathbf{G}\mathbf{H}:\dots:\mathbf{C}\mathbf{G}^{n-1}\mathbf{H}] = m \quad (2.64)$$

Cabe destacar que la existencia de la matriz \mathbf{D} en la ecuación de salida ayuda a establecer la controlabilidad del sistema. En el caso de que no existiera la matriz \mathbf{D} (salida $y(kT) = \mathbf{C}x(kT)$), la condición necesaria y suficiente sería

$$\text{Rango}[\mathbf{C}\mathbf{H}:\mathbf{C}\mathbf{G}\mathbf{H}:\dots:\mathbf{C}\mathbf{G}^{n-1}\mathbf{H}] = m \quad (2.65)$$

En este caso, la controlabilidad de la salida se produce si y sólo si los m renglones de \mathbf{C} son linealmente independientes.

Si se realiza un análisis de ambos casos estudiados, la forma en la que aparece \mathbf{D} tiene más probabilidades de ser controlable. Al introducirse una columna extra en la matriz de controlabilidad (la correspondiente a \mathbf{D}), se podría dar la situación en la que se pase de tener $m-1$ columnas linealmente independientes a m , por lo que se conseguiría la controlabilidad. Por simplificar, encontrar m vectores linealmente independientes es igual o más fácil entre $n+1$ vectores que en solo n vectores. [13]

2.8.3. Observabilidad

Para analizar la observabilidad de un sistema de control, se hará uso de un sistema LTI discreto descrito por

$$\begin{aligned} x((k+1)T) &= \mathbf{G}x(kT) \\ y(kT) &= \mathbf{C}x(kT) \end{aligned} \quad (2.66)$$

donde

- $\mathbf{x}(kT)$ = vector de estado (dimensión n) en el k -ésimo instante de muestreo
- $\mathbf{y}(kT)$ = vector de salida (dimensión m) en el k -ésimo instante de muestreo
- \mathbf{G} = matriz de $n \times n$
- \mathbf{C} = matriz de $m \times n$

El hecho de utilizar en este caso un sistema sin excitación es que, para la investigación de la condición necesaria y suficiente para la completa observabilidad, basta considerar el sistema 2.66, puesto que las matrices que se añaden a un sistema con variables de entrada son conocidas y solamente supondrían “complejidad” al estudio de las ecuaciones.

Se dice que un sistema es **completamente observable** si cualquier estado inicial $x(0)$ puede determinarse a partir de la “observación” de $y(kT)$ sobre un número finito de períodos de muestreo, es decir, si cualquier transición del estado de manera eventual afecta a todos los elementos del vector de salida. Otra posible definición sería: un sistema es completamente observable si cada variable de estado del sistema afecta alguna de las salidas.

La observabilidad es un fenómeno realmente útil para dar solución a un problema en el que las variables de estado de un sistema no son medibles puesto que dichas variables no son accesibles. La solución de la ecuación 2.66 será

$$\begin{aligned} x(kT) &= \mathbf{G}^k x(0) \implies \text{por tanto} \\ y(kT) &= \mathbf{C}\mathbf{G}^k x(0) \end{aligned} \quad (2.67)$$

La completa observabilidad implica lo siguiente: conocidos $y(0), y(T), y(T), \dots$, es posible determinar $x_1(0), x_2(0), \dots, x_n(0)$. Puesto que se necesitan encontrar n incógnitas, se necesitan únicamente n valores diferentes de $y(kT)$. Esto significa que se pueden utilizar los primeros n valores de $y(kT)$ ó $y(0), y(T), y(T), \dots, y((n-1)T)$ que permiten determinar $x_1(0), x_2(0), \dots, x_n(0)$. Es decir, dados

$$\begin{aligned} \mathbf{y}(0) &= C\mathbf{x}(0) \\ \mathbf{y}(T) &= CG\mathbf{x}(0) \\ &\vdots \\ \mathbf{y}((n-1)T) &= CG^{n-1}\mathbf{x}(0) \end{aligned}$$

se debe ser capaz de determinar $x_1(0), x_2(0), \dots, x_n(0)$. Realizando el análisis dimensional correspondiente, se puede observar que $y(kT)$ es un vector de dimensión m , y puesto que se tienen n ecuaciones, se llega a un sistema de $n \times m$ ecuaciones, todas ellas incluyendo $x_1(0), x_2(0), \dots, x_n(0)$. Si se fija como objetivo obtener una solución de $x_1(0), x_2(0), \dots, x_n(0)$ a partir de esas $n \times m$ ecuaciones, se debe imponer que exactamente n de ellas sean linealmente independientes, o lo que es lo mismo

$$\text{Rango}[C^*:G^*C^*:\dots:(G^*)^{n-1}C^*] = n \quad (2.68)$$

donde C^* significa la transpuesta conjugada de C . Se ha llegado, por tanto, a la condición necesaria y suficiente para que el sistema de las ecuaciones 2.66 sea completamente observable. La matriz de 2.68 se conoce como *matriz de observabilidad*. [13]

2.8.4. Principio de Dualidad

Ahora que se han definido los conceptos de controlabilidad y observabilidad, se analizará a continuación la relación que existe entre ambas definiciones. Para ello, se considerarán los dos sistemas siguientes

$$\begin{aligned} x((k+1)T) &= Gx(kT) + Hu(kT) \\ y(kT) &= Cx(kT) \end{aligned} \quad (2.69)$$

el cual será nombrado como S_1 . Para S_2 se tendrá lo siguiente

$$\begin{aligned} \hat{x}((k+1)T) &= G^*\hat{x}(kT) + C^*\hat{u}(kT) \\ \hat{y}(kT) &= H^*\hat{x}(kT) \end{aligned} \quad (2.70)$$

donde

- $\mathbf{x}(kT)$ y $\hat{\mathbf{x}}(kT)$ = vector de estado (dimensión n) en el k -ésimo instante de muestreo
- $\mathbf{u}(kT)$ y $\hat{\mathbf{u}}(kT)$ = vector de control (dimensión r) en el k -ésimo instante de muestreo
- $\mathbf{y}(kT)$ y $\hat{\mathbf{y}}(kT)$ = vector de salida (dimensión m) en el k -ésimo instante de muestreo
- \mathbf{G} = matriz de $n \times n$, \mathbf{G}^* = transpuesta conjugada de \mathbf{G}
- \mathbf{H} = matriz de $n \times r$, \mathbf{H}^* = transpuesta conjugada de \mathbf{H}

- \mathbf{C} = matriz de $m \times n$, \mathbf{C}^* = transpuesta conjugada de \mathbf{C}

La analogía que existe entre controlabilidad y observabilidad se denomina *Principio de dualidad*, y se conoce gracias a Kalman. Y dice así

“El sistema S_1 definido por las ecuaciones 2.69 es de estado completamente **controlable (observable)**, si y sólo si el sistema S_2 definido por las ecuaciones 2.70 es de estado completamente **observable (controlable)**”.

La demostración de este principio es bastante sencilla y se basa en las condiciones necesarias y suficientes para la controlabilidad y observabilidad completas. Si se escriben ambas condiciones, analizando ambos sistemas por separado, se comprenderá mejor

1. Controlabilidad S_1 : $\text{Rango}[\mathbf{H} : \mathbf{GH} : \dots : \mathbf{G}^{n-1}\mathbf{H}] = n$
2. Observabilidad S_1 : $\text{Rango}[\mathbf{C}^* : \mathbf{G}^*\mathbf{C}^* : \dots : (\mathbf{G}^*)^{n-1}\mathbf{C}^*] = n$
3. Controlabilidad S_2 : $\text{Rango}[\mathbf{C}^* : \mathbf{G}^*\mathbf{C}^* : \dots : (\mathbf{G}^*)^{n-1}\mathbf{C}^*] = n$
4. Observabilidad S_2 : $\text{Rango}[\mathbf{H} : \mathbf{GH} : \dots : \mathbf{G}^{n-1}\mathbf{H}] = n$

Como se puede observar comparando ambas condiciones, se evidencia la verdad del principio de dualidad. Mediante la utilización de este principio, la observabilidad de un sistema dado puede verificarse al probar la controlabilidad del estado de su dual. [13]

2.9. Transformación de un sistema en formas canónicas

Existen una serie de las llamadas *transformaciones de similitud* las cuales son realmente útiles a la hora de trabajar en el análisis y el diseño en el dominio de estado. Es posible que se quiera trabajar con este tipo de ecuaciones particulares por diversas razones, como se explicarán a continuación.

Se denominan transformaciones de similitud puesto que tanto el sistema de partida como el sistema transformado conservan las mismas ecuaciones características, vectores característicos, valores característicos y la función de transferencia. A continuación se describen las transformaciones de la Forma Canónica Controlable (FCC) y de la Forma Canónica Observable (FCO) a partir del siguiente sistema

$$\begin{aligned} x(k+1) &= Gx(k) + Hu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{2.71}$$

2.9.1. Forma Canónica Controlable

Considerese la matriz de transformación $P = SM$, donde

$$S = \begin{bmatrix} H & \vdots & GH & \vdots & G^2H & \vdots & \dots & \vdots & G^{n-1}H \end{bmatrix}, \quad M = \begin{bmatrix} a_1 & a_2 & \dots & a_{n-1} & 1 \\ a_2 & a_3 & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1} & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix} \tag{2.72}$$

donde \mathbf{S} es conocida como la **matriz de controlabilidad** y a_i son los coeficientes de la ecuación característica de G , la cual es

$$|sI - A| = s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0 = 0 \quad (2.73)$$

Definiendo el estado $x(k)$ a partir de la matriz de transformación P en función de otro estado $\hat{x}(k)$

$$x(k) = P\hat{x}(k) \quad (2.74)$$

Se puede afirmar que el sistema

$$\begin{aligned} \hat{x}(k+1) &= \hat{G}\hat{x}(k) + \hat{H}u(k) \\ y(k) &= \hat{C}\hat{x}(k) + \hat{D}u(k) \end{aligned} \quad (2.75)$$

se encuentra en Forma Canónica Controlable (FCC) si

$$\begin{aligned} \hat{G} = P^{-1}GP &= \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{bmatrix}, & \hat{H} = P^{-1}H &= \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \\ \hat{C} = CP, & & \hat{D} = D \end{aligned} \quad (2.76)$$

donde las matrices \hat{C} y \hat{D} no siguen ningún patrón en particular. Este tipo de transformaciones tiene un requisito indispensable: que la matriz P^{-1} exista, lo cual implica que la matriz S tenga inversa, puesto que la inversa de M siempre existe (su determinante es $(-1)^{n-1}$, nunca cero). [10] [13]

Esta forma de expresar en modo de espacio estados una función de transferencia garantiza que el sistema que se esté modelando sea **controlable**, término que se ha descrito con anterioridad. Cuando un sistema es controlable, es decir, se pueden modificar todos y cada uno de los estados mediante las entradas, se puede expresar mediante su FCC. En el caso del estudio de la trayectoria de un vehículo no tripulado, interesa que el modelo que simula el sistema sea controlable, y por tanto, se podría expresar según su FCC. Esa controlabilidad será la que permita al piloto modificar la velocidad, posición, actitud y cualquier otra variable definida como estado del sistema.

2.9.2. Forma Canónica Observable

Este método es bastante similar, y se le considera una forma dual de la transformación de la FCC. El sistema de 2.71 se transforma a la Forma Canónica Observable mediante la matriz Q

$$x(k) = Q\hat{x}(k), \quad Q = (MV)^{-1} \quad (2.77)$$

donde M viene definido por la ecuación 2.72, y

$$V = \begin{bmatrix} C \\ CG \\ CG^2 \\ \vdots \\ CG^{n-1} \end{bmatrix} \quad (2.78)$$

V se denomina comúnmente **matriz de observabilidad**, y debe cumplir el requisito de que V^{-1} debe existir para que la transformación FCO sea posible. Las distintas matrices involucradas en las ecuaciones transformadas quedarían de la siguiente forma

$$\hat{G} = Q^{-1}GQ = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{bmatrix}, \quad \hat{C} = CQ = [0 \quad 0 \quad \cdots \quad 0 \quad 1] \quad (2.79)$$

$$\hat{H} = Q^{-1}H, \quad \hat{D} = D$$

donde los elementos de las matrices \hat{H} y \hat{D} no están restringidos de ninguna forma, sino que dependerán de la función de transferencia de partida. Cabe destacar que si se analizan estas matrices de \hat{G} y \hat{C} se puede observar que equivalen a las traspuestas de las matrices \hat{G} y \hat{H} de la expresión 2.76, y por ello se le denominaba forma dual. [10] [13]

Para poder expresar un modelo mediante su FCO es necesario que el sistema bajo estudio sea completamente **observable**, es decir, que no haya salidas que no dependan de los estados directamente, o dicho de otra forma, que cada variable de estado afecte a alguna salida. La principal ventaja que poseen estas dos representaciones, FCC y FCO, son principalmente la rapidez con la que pueden ser calculadas. Una vez que se conozca que el sistema modelado cumple con los requisitos de controlabilidad u observabilidad, obtener el modelo de espacio estados a través de su ecuación diferencial o función de transferencia es inmediato, simplemente mirando los coeficientes. En el caso de no cumplir con dichos requisitos, sería necesario realizar cálculos más tediosos para obtener las matrices del modelo de espacio estados.

2.10. Descripción de un sistema en parte controlable/observable y no controlable/no observable

En la práctica, es posible que el problema bajo estudio no sea completamente controlable/observable, es decir, que tengan elementos del sistema que sí lo son pero otros no. Se verá en este apartado que es posible descomponer un sistema en parte controlable/no controlable y observable/no observable. Esta idea resulta de utilidad para un mejor análisis de los datos de un sistema, pudiendo resolver la parte controlable/observable por un método y estudiar minuciosamente la parte restante del sistema.

Se considerarán las mismas ecuaciones de estado que se han venido estudiando hasta el momento, como las de la expresión 2.71.

2.10.1. Parte controlable/no controlable

Una condición que se presupone para que en un sistema exista parte no controlable, es que su rango sea menor que la dimensión de la ecuación de estado, como se demostró en las condiciones necesarias y suficientes para la controlabilidad. Es decir

$$\text{Rango} [H \quad GH \quad \cdots \quad G^{n-1}H] = n_1 < n \quad (2.80)$$

De nuevo se usará la matriz P como matriz de transformación o cambio de coordenadas, de dimensión n , definida como

$$P^{-1} = [p_1 \quad p_2 \quad \cdots \quad p_{n_1} \quad \cdots \quad p_n] \quad (2.81)$$

donde las n_1 primeras columnas son linealmente independientes para que se cumpla la condición de la expresión 2.80. El resto se eligen arbitrariamente con el objetivo de hacer P una matriz no singular. Esta transformación ($\bar{x} = Px$) lleva al sistema a la siguiente expresión

$$\begin{aligned} \begin{bmatrix} \dot{\bar{x}}_C \\ \dot{\bar{x}}_{\bar{C}} \end{bmatrix} &= \begin{bmatrix} \bar{G}_C & \bar{G}_{12} \\ 0 & \bar{G}_{\bar{C}} \end{bmatrix} \begin{bmatrix} x_C \\ x_{\bar{C}} \end{bmatrix} + \begin{bmatrix} \bar{H}_C \\ 0 \end{bmatrix} u \\ y &= [\bar{C}_C \quad \bar{C}_{\bar{C}}] x + Du \end{aligned} \tag{2.82}$$

Los estados en las nuevas coordenadas se descomponen en

- \bar{x}_C : n_1 estados **controlables**
- $\bar{x}_{\bar{C}}$: $n - n_1$ estados **no controlables**

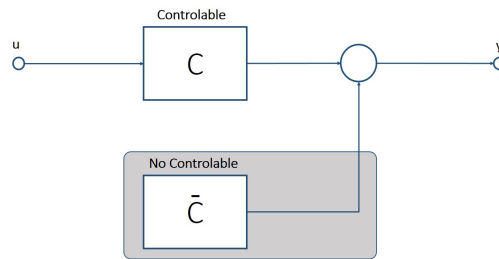


Figura 2.6: Descomposición parte controlable y no controlable

La ecuación de estado de orden reducido (puesto que se le eliminan aquellos elementos no controlables) de los estados controlables quedaría

$$\begin{aligned} \dot{\bar{x}}_C &= \bar{G}_C \bar{x}_C + \bar{H}_C u \\ \bar{y} &= \bar{C}_C \bar{x} + Du \end{aligned} \tag{2.83}$$

el cual es controlable y tiene la misma función de transferencia que el estado original. Como dato interesante, la función de Matlab *ctrbf* transforma una ecuación de estado en su forma canónica controlable/no controlable.

2.10.2. Parte observable/no observable

Este fenómeno es bastante parecido al anterior, pero tiene algunas diferencias debido a la dualidad mencionada previamente. En este caso, se supondrá que el rango será

$$\text{Rango} \begin{bmatrix} C \\ CG \\ \vdots \\ CG^{n-1} \end{bmatrix} = n_2 < n, \quad P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{n_2} \\ \vdots \\ p_n \end{bmatrix} \tag{2.84}$$

donde P vuelve a ser la matriz para cambiar de coordenadas y sus primeras n_2 filas son linealmente independientes, y el resto se eligen arbitrariamente para que no sea singular. De nuevo, con la transformación $\bar{x} = Px$ el sistema quedaría

$$\begin{aligned} \begin{bmatrix} \dot{\bar{x}}_O \\ \dot{\bar{x}}_{\bar{O}} \end{bmatrix} &= \begin{bmatrix} \bar{G}_O & 0 \\ \bar{G}_{21} & \bar{G}_{\bar{O}} \end{bmatrix} \begin{bmatrix} \bar{x}_O \\ \bar{x}_{\bar{O}} \end{bmatrix} + \begin{bmatrix} \bar{H}_O \\ H_{\bar{O}} \end{bmatrix} u \\ y &= [\bar{C}_O \quad 0] \bar{x} + Du \end{aligned} \quad (2.85)$$

Los estados en las nuevas coordenadas se descomponen en

- \bar{x}_O : n_2 estados **observables**
- $\bar{x}_{\bar{O}}$: $n - n_2$ estados **no observables**

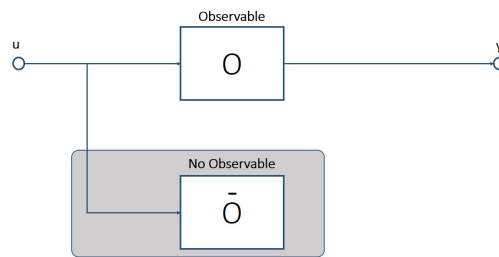


Figura 2.7: Descomposición parte observable y no observable

La ecuación de estado de orden reducido (puesto que se le eliminan aquellos elementos no observables) de los estados observables quedaría

$$\begin{aligned} \dot{\bar{x}}_O &= \bar{G}_O \bar{x}_O + \bar{H}_O u \\ \bar{y} &= \bar{C}_O \bar{x} + Du \end{aligned} \quad (2.86)$$

el cual es observable y tiene la misma función de transferencia que el estado original. Como dato interesante, la función de Matlab *obsvf* transforma una ecuación de estado en su forma canónica observable/no observable.

3 | Modelado de Sensores en los Vehículos Aéreos no Tripulados

Para adaptarse a su entorno, los seres humanos disponen de un sistema de sentidos, los cuales son clave para conocer el entorno que los rodea. Los UAV son máquinas que necesitan de un propio sistema de sensores que les permitan saber dónde están, cómo es ese lugar en el que se encuentran, en qué condiciones físicas se encuentran, dónde están los objetos a su alrededor, etc. Existen una amplia gama de dispositivos diseñados para recibir datos externos de magnitudes físicas, y transformarlas en valores eléctricos, los cuales se pueden introducir a un sistema de control. Es decir, un sensor es aquel dispositivo sensible a magnitudes físicas, y debe ser capaz de transformar dicha magnitud física para que pueda ser utilizada por los demás sistemas que conforman el UAV.

Las tareas cotidianas que realiza un ser humano implican una enorme complejidad para implementarlas en sistemas robóticos. Es por ello necesario disponer de un conocimiento minucioso de los sistemas sensoriales que se pueden aplicar a la robótica y a sistemas automáticos, como son los UAVs, para así conocer cuál de ellos es el más adecuado para desarrollar una tarea concreta.

Un sensor es un dispositivo eléctrico y/o mecánico que convierte magnitudes físicas (como pueden ser luz, magnetismo, calor, presión, altitud, movimiento...) a valores medibles de dicha magnitud. Por ejemplo, existen dispositivos que transforman una acción de presión (modificando consecuentemente su tamaño), en un cambio de su resistencia. Los sensores aportan información valiosa tanto del entorno de trabajo del UAV como de su estado interno para que se realice la tarea de manera eficiente.

Se realizará un análisis en este apartado sobre los diferentes tipos de sensores que existen, y las características de cada uno para así decidir de manera óptima cuál será el adecuado para realizar una determinada tarea. La magnitud a medir será un factor decisivo a la hora de elegir un sensor u otro para una aplicación. También serán importantes, aparte de las características del fenómeno a medir (velocidad, presión, desplazamiento...), otras características como el rango de la medida, dinámica del sensor, tipo de señal de salida (condicionará el procesamiento de la salida), coste, precisión y resolución requerida. La información que aporta el sensor debe ser tratada y procesada, y normalmente este proceso se realiza en tres fases:

- El sensor mide el fenómeno físico. Como consecuencia, muestra en su salida una señal dependiente del valor de la lectura.
- La señal de salida es modificada por un sistema de acondicionamiento de la señal, cuya salida es un voltaje que será convertido, mediante convertor A/D (analógico/digital), en una señal tratable por un ordenador.
- Normalmente, los sensores disponen de circuitos internos encargados de modificar o amplificar la tensión de salida, la cual pasa al convertidor A/D y de ahí al PC. Dicho convertidor A/D usualmente sólo es capaz de medir valores dentro de un rango determinado de tensiones, normalmente entre 0 y 5V. El uso de este dispositivo permite que la salida continua se traduzca a niveles discretos.

Aquí se puede observar cómo en el uso de sensores en el ámbito de los UAVs se distinguen varias disciplinas diferentes a la Aeroespacial, como pueden ser la informática, para el procesamiento de la información sensorial; la teoría de la señal, para el acondicionamiento de la señal originada por el sensor; la electrónica, en la circuitería interna del mismo, etc. [22]

Los UAVs han experimentado una atención y evolución significativas en los últimos años. Puesto que su uso cada vez se extiende a más campos fuera de la Aeronáutica, es necesario implementar una serie de sistemas mediante los avances en tecnología que sean capaz de realzar su fiabilidad, autonomía y estabilidad. De esta forma se aumentará la percepción que el dron puede llegar a tener sobre su estado, y así poder auto-adaptarse a las condiciones requeridas.

Los drones están normalmente equipados con sistemas micro-electro-mecánicos *MEMS* (del inglés: *micro-electro-mechanical systems*) de bajo coste y peso como son las Unidades Inerciales de Medida *IMU* (del inglés: *Inertial Measurement Units*) como se detallarán más adelante. Estas suelen tener un giróscopo, acelerómetro y magnetómetro de tres ejes, los cuales son susceptibles a fallos debidos a daños en los componentes, variaciones de temperatura o excesiva vibración (entre otros). Puesto que la dinámica de la mayoría de estos UAVs se rige por un modelado no lineal, muchos investigadores han desarrollado soluciones a los fallos producidos basándose en la linealización de los modelos alrededor de algún punto de equilibrio, como se ha analizado en la sección 2 (Modelado en el Espacio de Estados, Linealización de las ecuaciones de estado). [2]

3.1. Caracterización de los Sensores

En esta sección se van a explicar diferentes propiedades que le dan una identidad propia a cada sensor. Se considera bastante importante conocer este tipo de parámetros, puesto que de ellos va a depender en gran parte el desarrollo y los resultados de cada caso. Por ejemplo, el utilizar sensores en los que no se conozca su resolución puede llevar a desconcierto a la hora de analizar los errores en el resultado, sin saber su procedencia. Para conseguir que el UAV realice la tarea para la cual fue diseñado con precisión es importante tanto que él mismo tenga conocimiento de su estado y del entorno, como de que el diseñador tenga un buen conocimiento de los parámetros que definen cada componente que lo forma.

Dentro de la amplia variedad de sensores que existen en el mercado, estos se pueden clasificar en dos grandes grupos: internos y externos.

- **Sensores Internos o Propioceptivos.** Son aquellos que se encuentran integrados en la propia estructura del UAV, y aportan información sobre valores de parámetros internos, como pueden ser posición, velocidad o ángulo de giro de cada una de las hélices. Son imprescindibles en la mayoría de las aplicaciones, puesto que sirven, por ejemplo, para indicar al vehículo cuando es el momento adecuado de recargar las baterías, cuando algún dispositivo se está sobrecalentando o cuando algún componente no está funcionando de la manera adecuada.
- **Sensores Externos o Exteroceptivos.** Se encuentran ubicados fuera de la estructura, y aportan información del entorno en el que se encuentra el dron, como puede ser la proximidad, reconocimiento o localización de un objeto, el contacto, etc. Típicamente se ayudan de una cámara mediante la cual el controlador es capaz de guiar el vehículo e identificar y/o manipular objetos de su entorno.

Para definir las propiedades que caracterizan a un sensor, se utilizan los llamados descriptores. Estos descriptores se pueden clasificar en estáticos, aquellos que definen el comportamiento en régimen permanente de un sensor, o dinámicos, aquellos que caracterizan la evolución temporal de la señal de salida ante determinadas señales de excitación a la entrada.

3.1.1. Descriptores estáticos

- **Rango.** Valores máximos y mínimos para las variables de entrada y salida. Se suele proporcionar como el rango de la entrada física que el sensor puede convertir en señales eléctricas.

- **Exactitud.** Normalmente, el valor de la lectura de un sensor no es un dato preciso, sino que se trata de una distribución estadística alrededor del valor real. La exactitud sería la desviación de dicha lectura con respecto a una entrada conocida. Se define como el mayor error esperado entre señal medida e ideal, normalmente en forma de porcentaje.
- **Repetitividad.** Capacidad de reproducir una lectura con una precisión dada. Se expresa también en porcentaje respecto al valor máximo de rango de funcionamiento.
- **Reproducibilidad.** Es la misma propiedad que la repetitividad, excepto que se usa solamente cuando distintas mediciones se toman bajo diferentes condiciones.
- **Resolución.** Normalmente, la salida respecto a la entrada se representa como una función continua. Pero es posible que la salida varíe siguiendo unos pequeños saltos con respecto a la entrada continua. Pues bien, la resolución es la cantidad más pequeña de ese incremento que puede ser determinada. Por ejemplo, un termómetro tiene una resolución de 1° , y hacen 25.4° . El termómetro marcará 25° , puesto que su resolución no le permite dar valores decimales.
- **Error.** Diferencia entre el valor medido por el dispositivo y el real.
- **No linealidad.** El comportamiento de un sensor sigue un modelo no lineal, pero en ocasiones su evolución se supone lineal respecto a la variable de entrada con el fin de facilitar los cálculos. Pero este hecho conlleva errores por no linealidades, puesto que la señal sufre desviaciones con respecto a la salida teórica lineal. En la siguiente Figura 3.1 se puede apreciar que la función $N(x)$ muestra los errores que se producen por la diferencia existente entre la respuesta del sensor y la función lineal por la que es aproximado $f(x)$.

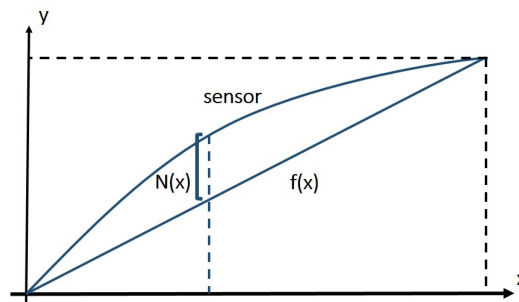


Figura 3.1: Respuesta no lineal de un sensor [22]

Dentro de las no linealidades, tres de las más frecuentes se deben a

1. **Saturación.** Se produce cuando una vez pasado un valor determinado de la entrada, el dispositivo deja de aumentar linealmente y se estanca en dicho *valor de saturación*. Figura 3.2 a)
2. **Zona muerta.** Se define como el rango de las entradas en el que el dispositivo deja de tener comportamiento lineal. En la Figura 3.2 b) se puede observar la zona desde que el sensor deja de responder hasta que comienza a mostrar valores de salida de nuevo adecuados. Un ejemplo típico de este comportamiento lo presentan los amplificadores de potencia.
3. **Histéresis.** Este es un tipo de no linealidad común entre los transformadores eléctricos. Como se muestra en la Figura 3.2 c) el valor de la medida estará sujeta al momento en el que ésta se realice.

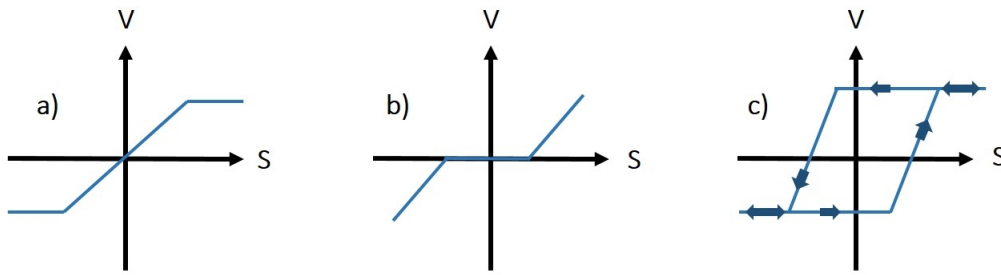


Figura 3.2: Tipos de no linealidades [22]

- **Sensibilidad.** Proporción de cambio de la salida frente a cambios en la entrada. Es común representarlo como variaciones de voltaje frente a variaciones en la entrada, $\partial V/\partial x$. Es, en definitiva, una medida que indica hasta qué punto la señal eléctrica de salida es capaz de cambiar tal y como cambian las magnitudes de entrada. Si se considera que la salida del sensor es V y la magnitud física es x , la sensibilidad del sensor se define según el factor de sensibilidad S como sigue

$$\frac{\Delta V}{V} = S \frac{\Delta x}{x} \quad (3.1)$$

- **Excitación.** Cantidad de corriente o voltaje mínima requerida para que funcione/se active el sensor.
- **Estabilidad.** Es la posibilidad que tiene un sensor de mostrar la misma salida en un rango en el que la entrada permanece constante.
- **Ruido.** La señal de salida medida típicamente no se corresponde con la ideal, puesto que existe un ruido que la modifica. Este fenómeno causa problemas a los procesadores que la interpretan. Su previo conocimiento será vital para mitigar sus efectos negativos. [22]

3.1.2. Descriptores dinámicos

- **Tiempo de retardo, t_d .** Tiempo que tarda la señal de salida en alcanzar el 50 por ciento del valor final.
- **Tiempo de subida, t_r .** Tiempo que tarda la señal de salida en alcanzar el valor final por primera vez, y se encuentra íntimamente relacionado con la velocidad de respuesta del sensor.
- **Tiempo de pico, t_p .** Tiempo que tarda la señal de salida en alcanzar el pico máximo de sobreoscilación.
- **Pico de sobreoscilación, M_s .** Expresa la variación máxima que se produce en la salida con respecto al valor final, y se suele medir en porcentaje. En otras palabras, es la amplitud que posee el primer pico.
- **Tiempo de establecimiento, t_s .** Tiempo que tarda la salida del sensor en alcanzar una banda de 5 por ciento en torno al valor final, y se estabiliza alrededor de ésta.

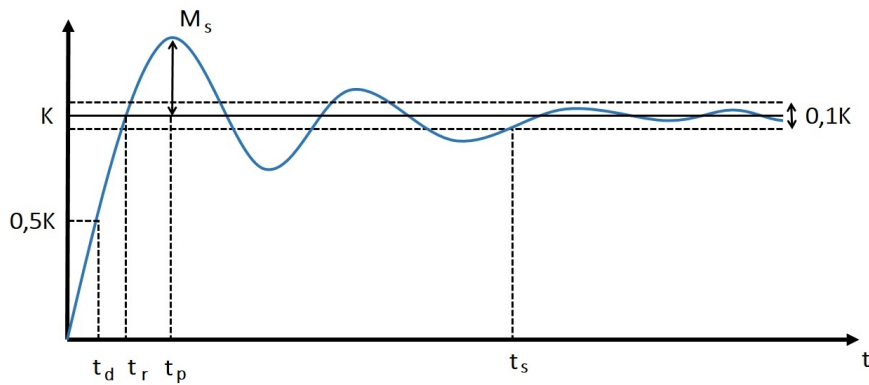


Figura 3.3: Descriptores dinámicos en la respuesta de un sensor [22]

3.2. Uso correcto de los sensores

Los continuos errores en las actuaciones de todo tipo de sistemas, ya no solamente en el ámbito bajo estudio de los UAVs, debido a una falta de comprobación de los equipos utilizados están a la orden del día. En parte se puede argumentar debido al moderado proceso de globalización, en el que investigadores de por ejemplo Europa encargan dispositivos procedentes de Asia o América, donde existen unos patrones, normativas y estándares no siempre equivalentes a los suyos. Es necesario realizar una serie de análisis y algunos tests con el fin de comprobar que la utilización de dichos dispositivos se ajusta de forma adecuada a lo esperado. Llevar a cabo este tipo de controles evitará pérdidas de tiempo, y sus costes implicados, a la hora de comprobar la instalación completa donde ya no sólo actúan sensores sino todo tipo de sistemas.

En las hojas de características de cada sensor aparecen una serie de factores, aparte de los descriptores estáticos y dinámicos, que pueden afectar de manera crucial a su funcionamiento. Ejemplos de estos factores pueden ser condiciones ambientales como vibraciones, humedad o rango de temperaturas en el que se garantiza el correcto funcionamiento. En el uso de UAVs, es bastante más influyente este tipo de fenómenos que en sistemas que se encuentren en robots utilizados en el interior de naves industriales. En la realidad se utilizan los drones al aire libre, bajo los efectos de vibraciones debidas al posible viento, variaciones importantes de humedad si se encuentra sobrevolando zonas de ríos, pantanos o mares, y variaciones de temperatura si la altura alcanzada es considerable o si está operando en zonas montañosas.

Se describirán a continuación una serie de procesos recomendables previos a la integración de los sensores que se explicarán a continuación, con el fin de optimizar su uso.

3.2.1. Calibración de sensores

Las respuestas que proporcionan los sensores son conocidas e interpretables por los usuarios o por el resto de sistemas que dependen de él. El proceso de calibración se basa en comparar la respuesta obtenida por el sensor con otros que tengan una respuesta estándar conocida. Mediante este proceso, se define la relación entre la variable medida por el sensor y su señal de salida.

Los UAVs normalmente son diseñados con una inestabilidad intrínseca, y los sistemas que lo forman solo pueden ser estabilizados gracias a los sensores de alta precisión. Con el objeto de conseguir una actuación del dron óptima, es necesario que la medida proporcionada por el sensor sea lo más precisa posible.

Uno de los sensores más típicos que están incorporados en estos vehículos son las **Unidades de Medida Inercial**, comúnmente conocidas como **IMUs** (*Inertial Measurement Units*), las cuales miden aceleraciones lineales y velocidades angulares en 3 dimensiones, como se explicará a continuación. La calibración de una IMU está considerada como un gran reto dentro de la navegación inercial. Aunque

existen IMUs altamente fiables en el mercado, **es aconsejable realizar calibraciones independientes** con el fin de verificarlas. También se recomienda calibrar la IMU una vez que se encuentre embarcada en el UAV, puesto que la localización y orientación relativa del sensor con respecto al cuerpo del UAV puede ser considerada fuente de errores si no se ha modelado correctamente, es decir, si se ha colocado con una determinada orientación pero luego no se considera esa misma orientación para implementar los sistemas de control del dispositivo. Para facilitar los cálculos en la actitud de la IMU, normalmente se coloca en el centro de gravedad del UAV, y además, se alinean los ejes del sensor con su estructura (la del UAV). Algunos drones, especialmente cuatrirrotores, tienen un espacio disponible bastante limitado, y puede resultar complicado colocar exactamente el sensor en la posición deseada. En [26], se propone realizar el proceso de calibración y verificación de una IMU ya embarcada en el vehículo a partir de un sistema “*hexapod*” (6 patas) con 6 grados de libertad, para poder comprobar todos los movimientos posibles en todas las direcciones.

Una aeronave suele tener numerosas **superficies de control**, puesto que garantizan la estabilidad del vehículo. En el caso de los vehículos aéreos no tripulados, el diseño de las superficies de control es una combinación bastante compleja de un gran número de disciplinas (mecánica, aviónica, electrónica, diseño, control etc.) que se interconectan para obtener la configuración óptima. Debido a que las superficies de control son las encargadas de la buena maniobrabilidad y controlabilidad (despegue, subida, vuelo en parado, bajada, aterrizaje, etc.) del UAV, es necesaria una buena implementación/diseño de las mismas. Así, se podrán alcanzar los requisitos necesarios para adquirir el certificado de uso comercial o militar. Con el uso de estos vehículos, es posible que se produzcan errores en la calibración de estas superficies, por diferentes motivos: desplazamientos dentro de la estructura por algún golpe, mal funcionamiento por daños, roturas, daños de componentes por temperaturas extremas, etc. [18] [22]

Algunos estudios afirman que para la calibración de sensores como pueden ser **magnetómetros** (efecto Hall, brújulas electrónicas, interruptores magnéticos, etc.) es necesario realizarla en un área a cielo abierto. Este hecho se debe a que para determinar el norte magnético terrestre, es necesario estar alejados de elementos ferromagnéticos, puesto que el campo magnético varía notablemente en sus cercanías. También es posible que se de el caso en el que se tengan distintos sensores que proporcionen información redundante (por ejemplo, un barómetro y un ultrasonido para medir alturas). En este caso, sería conveniente realizar dos tipos de calibraciones, dado que los errores y sensibilidades que presentan cada uno son muy diferentes. [9]

3.2.2. Modelado de simulaciones SIL-HIL - *Software-in-the-loop* y *Hardware-in-the-loop*

Las técnicas SIL-HIL son utilizadas para el desarrollo y comprobación de sistemas embebidos en tiempo real. Son usualmente conocidas como un método utilizado para desarrollar prototipos de sistemas de control y aplicaciones de modelado, como el caso bajo estudio. En el caso de los UAVs, se utilizan para probar los sistemas embarcados antes de su utilización real, para detectar así cualquier tipo de anomalía y corregirla a tiempo.

Las referencias internacionales definen **HIL** como un avance en el ámbito de la ingeniería en general, usado en el desarrollo y pruebas de sistemas embebidos complejos a tiempo real, reuniendo ventajas tanto para el prototipo físico como para el virtual. Es una técnica que combina el modelo de **simulación** matemático de un sistema con el **hardware físico real**, mediante una serie de interacciones bidireccionales (de la simulación al hardware y viceversa). La ingeniería moderna requiere el uso de sistemas con alto grado de modelado de prototipos y HIL se ha convertido en una técnica indispensable en muchas aplicaciones, no solamente en el ámbito aeroespacial (control de sistemas de propulsión), sino también automatización industrial (sistemas de seguridad para automóviles), plantas nucleares, sistemas automotrices, la marina y aplicaciones militares (sistemas de defensa), entre otras. [21] [7]

Esta técnica de simulación está evolucionando rápidamente, puesto que la creación de prototipos virtuales de un sistema puede apoyar al trabajo inicial de diseño de dicho sistema de forma **rápida** y **eficiente**. Las principales ventajas de utilizar técnicas de modelado HIL son las siguientes [7]:

- **Rentabilidad y prototipado.** Requiere menos hardware en comparación con los prototipos físicos, de modo que supone menor coste y son más rápidos de construir.
- **Fidelidad y verosimilitud.** Alcanzan niveles de fidelidad inalcanzables por simulaciones puramente virtuales, puesto que estos no son capaces de reproducir completamente algunas de las características o atributos del sistema real.
- **Velocidad de simulación.** Alcanzan mayor velocidad que las simulaciones virtuales de los mismos fenómenos (por ejemplo, simulaciones de motor IC basados en Dinámica de Fluidos Computacional).
- **Repetitividad.** Aquellos sistemas que normalmente operan en entornos muy variables (por ejemplo, los sistemas de suspensión de vehículos fuera de la carretera) a menudo pueden ser probados en entornos de laboratorio controlados a través de la simulación HIL, que pueden aumentar significativamente la repetibilidad.
- **No altera la naturaleza.** Hace posible la simulación de eventos destructivos (accidentes de vehículos, interceptación de misiles, etc.) sin incurrir en una destrucción real, y por lo tanto, costosa.
- **Integralidad.** Hace posible la simulación de un sistema en un rango mucho más amplio de sus condiciones de funcionamiento posibles, a través de la creación de prototipos puramente físicos.
- **Seguridad.** Se pueden utilizar para entrenar a los operadores humanos (por ejemplo, pilotos de avión) o a sistemas críticos para la seguridad (por ejemplo, aviones supersónicos) en entornos significativamente más seguros (por ejemplo, simuladores de vuelo).
- **Coexistencia de sistemas de ingeniería.** Permite que diferentes equipos desarrollen las diferentes partes de un sistema en el hardware sin perder de vista los problemas de integración, permitiendo de este modo la ingeniería de sistemas concurrentes.

Por otra parte, el sistema **SIL** corren también en tiempo real con **RTOS** (del inglés, *Real Time Operative Systems*) y sirven para desarrollar leyes de control y de navegación, donde todo se simula por *software* (tanto los **sensores** como los actuadores). En el caso de los vehículos aéreos no tripulados, el simulador SIL podría consistir en varios ordenadores en los que se simula la GCS (del inglés, *Ground Control Station*), y un ordenador en el que se simula el modelo de aeronave (modelo dinámico, aerodinámico, de masa e inercial, flight control: navegación-guiado-control, el sistema de gestión de la misión, generación y envío de las medidas tomadas por los sensores y actuadores, etc.).

En cualquier caso, ambas técnicas son utilizadas para llevar a cabo experimentos que sean lo más similares posibles a las actuaciones reales que tendrán que realizar los sistemas embarcados. De esta forma, se pueden analizar los fallos o carencias que posean los dispositivos antes de ponerlos en funcionamiento, por lo que supondrá un ahorro considerable en el diseño de sistemas.

3.3. Sensores para medir distancias y proximidad

Se describen este tipo de sensores en primer lugar debido a su importancia. Dentro de las distintas funcionalidades que puede tener un UAV, en la mayoría de ellas será necesario conocer los objetos que lo rodean para realizar una función concreta. Por ejemplo, en campañas anti-incendios se necesita conocer el área invadida por las llamas, en control de plagas en agricultura es necesario cubrir todo el terreno con los correspondientes agentes plaguicidas, en mensajería aérea también es preciso conocer el punto exacto de entrega, etc. Estos sensores de proximidad surgen de esta necesidad, de conocer la posición en la que no existe contacto entre el actuador y el detector.

Para todas aquellas funciones, existen los sensores que miden la proximidad y la presencia de objetos situados a una distancia máxima de alcance del sensor. Se describirán en primer lugar sensores

que hacen uso de características eléctricas (capacitivos); después se explicarán aquellos que usan propiedades eléctricas y magnéticas (inductivos, basados en efecto Hall); y por último los que hacen uso de propiedades ópticas y ultrasonidos.

3.3.1. Sensores capacitivos

Cabe destacar en primer lugar la diferencia existente entre sensores capacitivos e inductivos. Los capacitivos funcionan detectando los cambios que se producen en la capacidad parásita que se origina entre el detector y objeto que se desea medir. Son utilizados para medir la distancia a objetos metálicos y no metálicos, como pueden ser la madera, líquidos y plásticos. Por su parte, los sensores inductivos están basados en el amortiguamiento que se produce en el campo magnético a causa de corrientes inducidas (o corrientes de Foucault) de los materiales situados alrededor del sensor. En este caso el material debe ser metálico para su funcionamiento.

Los sensores capacitivos se basan en el esquema representado en la Figura 3.4. Se observan condensadores genéricos, en los cuales se dispone de dos placas metálicas llamadas armaduras, separadas por un dieléctrico.

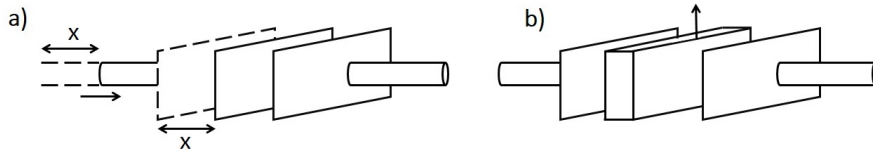


Figura 3.4: Esquemas de sensores capacitivos [22]

Los elementos capacitivos utilizados como condensadores son variables, de forma que el **desplazamiento a medir** provocará un **desplazamiento en algún componente** del condensador, y por lo tanto, una **modificación de su capacidad**. En la Figura 3.4 se observa cómo al acercarse un objeto se desplaza un componente en el condensador y por lo tanto, se modifica la capacidad. Esa variación permite medir el desplazamiento sufrido o la cercanía con el objeto.

Entre las dos placas se almacena una electricidad, la cual puede verse variada por la modificación de la posición del dieléctrico o la disposición de las mismas. En el caso de placas paralelas entre sí, el campo eléctrico creado entre ellas es uniforme, y la diferencia de potencial es

$$V_R = \frac{1}{\epsilon_a} \frac{Qd}{S} \quad (3.2)$$

donde Q es la carga de cada lámina, S el área, ϵ_a la constante dieléctrica del medio, y d la separación. La capacidad del condensador es la siguiente

$$C = \epsilon_a \frac{S}{d} \quad (3.3)$$

El cálculo de la distancia a medir se basa en el principio de que el potencial que se almacena es inversamente proporcional a la distancia que separa las placas del condensador. La cercanía con el objeto provocará un aumento de la capacitancia. Dicho aumento, mediante el proceso de calibrado conveniente, es traducido a una distancia como señal de salida del sensor. Para medir distancias mayores a algunos milímetros, la sensibilidad de estos sensores disminuye notablemente. En la Figura 3.5 se puede observar cómo la cercanía de un objeto, sea o no conductor, implica un aumento del campo eléctrico.

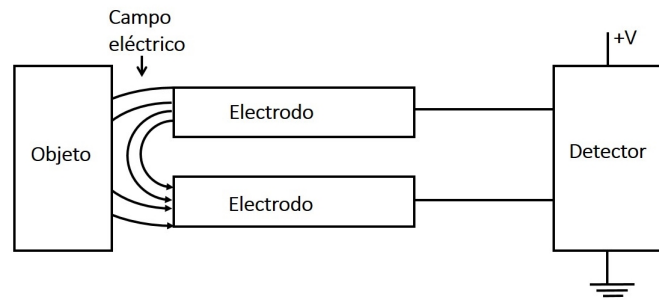


Figura 3.5: Esquema de sensor capacitivo como detector de proximidad

Una forma fácil de simplificar este funcionamiento sería: un cambio en la distancia se traduce en un desplazamiento en algún componente que provoca cambio de capacidad, mediante la variación del campo eléctrico, y esa capacidad es traducida a un cambio en la salida del sensor para su posterior interpretación.

3.3.2. Sensores inductivos

Para utilizar este tipo de sensores es necesario que el objeto del cual se quiere medir su distancia sea ferromagnético. Los sensores poseen una bobina y un imán permanente. Al colocar el imán cerca de la bobina y cambiar de posición el objeto ferromagnético, entre ambos componentes se producirá una variación del flujo magnético a través de la bobina, lo que inducirá una fuerza electromotriz. Esta corriente es la que se utilizará como detector de presencia del objeto a medir.

Normalmente son utilizados cuando el desplazamiento relativo entre el sensor y el objeto es lineal. Un ejemplo de sensores que utilizan este principio son los llamados *transformadores diferenciales*. Se dispone de una bobina primaria central y dos bobinas secundarias laterales enrolladas sobre un núcleo magnético, de forma que los desplazamientos que van a medir estos sensores van a ser los de este núcleo. Se proporciona una intensidad y tensión conocidas a la primaria, la cual es transformada en las bobinas secundarias en intensidades y tensiones diferentes.

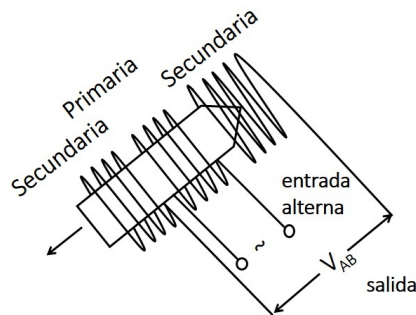


Figura 3.6: Ejemplo de transformador diferencial [22]

En la Figura 3.6 se puede observar el funcionamiento del transformador diferencial. El desplazamiento producido en el núcleo magnético se produce de tal forma que alguna de las bobinas secundarias no cubra completamente el núcleo. Como consecuencia de esto, la corriente inducida en una secundaria será mayor que en la otra, y a partir de esta diferencia, se podrá medir el desplazamiento sufrido por el núcleo. Con el núcleo centrado, la salida será de $0V$, mientras que si se desplaza a algún lado habrá más o menos corriente inducida en una de las dos secundarias, la cual se traducirá en distancia de desplazamiento.



Figura 3.7: Sensores capacitivos e inductivos reales (cortesía de Siriorred)

Este tipo de sensores, a pesar de no poder medir grandes desplazamientos, tienen ventajas como su rápida respuesta, alta resolución, linealidad, baja histéresis y repetitividad.

3.3.3. Basados en efecto Hall

Para utilizar un sensor que se base en este principio, es necesario que el objeto a medir sea capaz de producir un campo magnético. Se obtendrá una referencia de su cercanía o del desplazamiento que ha sufrido a partir de la diferencia de potencial generada como consecuencia del efecto Hall. Esta diferencia de potencial será mayor cuanto más intenso sea el campo magnético o más próximo se encuentre el objeto a medir.

¿Cómo funcionan realmente este tipo de sensores? Utilizando la presencia de un campo magnético en un semiconductor para producir cambios en la corriente eléctrica generada. Se utiliza esta presencia o ausencia del campo magnético para proporcionar un determinado nivel de tensión V , como se muestra en la Figura 3.8. Se observa que el campo magnético es perpendicular a la corriente eléctrica que atraviesa la placa conductora. Se genera así el campo eléctrico debido a la polarización de la placa en lado positivo y negativo, compensando el campo magnético.

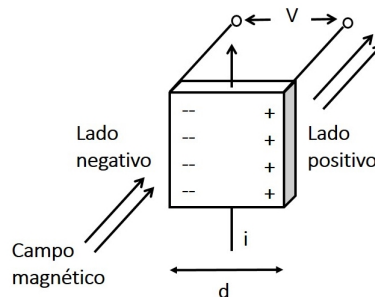


Figura 3.8: Efecto Hall [22]

Entre ambos extremos de la placa se genera la siguiente diferencia de potencial

$$V = K_H \frac{B_f i}{d} \quad (3.4)$$

donde K_H es el coeficiente de Hall, B_f es la densidad del flujo magnético, i es la intensidad de corriente y d el grosor de la placa. Típicamente este tipo de sensores se encuentran instalados en semiconductores. La electrónica integrada que poseen les permite proporcionar una señal que se encuentra amplificada y condicionada, lo cual supone una actuación más directa, fácil y económica.

En [5], proyecto en el que se intenta mejorar el sistema anemométrico y de monitorización energética en un UAV (Proyecto Céfireo del Departamento de Aeroespacial de la Universidad de Sevilla), se utilizan este tipo de sensores basados en el efecto Hall. En dicho proyecto, se contaban los pasos de las bobinas de un motor eléctrico por un punto estático mediante un sensor de efecto Hall, para la medición del régimen de giro del motor. En este caso se utilizó un *Allegro 1120 EUA-T*. Dicho sensor actuaba como interruptor, el cual se accionaba debido al paso del campo magnético generado por cada bobina del motor, puesto que esta elevaba la tensión terminal del sensor de $0V$ a $5V$. De esta forma era capaz de contar el tiempo entre vueltas, y así, la velocidad de giro del motor.

3.3.4. Basados en ultrasonidos

Los ultrasonidos son ondas con la capacidad de ser reflejadas si encuentran en su camino una discontinuidad o algún elemento extraño en el medio en el que viajan. Son prácticamente iguales a las ondas sonoras, por lo que viajan a una velocidad de 340m/s, pero poseen una mayor frecuencia (en torno a los 20kHz, por encima del umbral del oído humano, 16kHz). Su funcionamiento es “sencillo”: el emisor lanza un tren de pulsos ultrasónicos y espera el rebote en el receptor, para medir el tiempo transcurrido para así calcular la distancia a la que se encuentra. La reflexión de la onda en el objeto a medir se produce debido a la diferencia de impedancias acústicas entre el medio y el objeto.

El rango de actuación, en cuanto a distancia que pueden medir estos sensores, es mayor del que poseen los presentados con anterioridad. Pero también cabe destacar que presentan una serie de problemas de implementación bastante comunes, los cuales dificultan las buenas mediciones de distancias lejanas. Algunos de los problemas más comunes son:

- **Ángulo de incidencia.** La dirección del reflejo de la onda depende directamente de este ángulo. Cuanto menor sea, más probable será que no se detecte el eco o que se produzcan medidas erróneas.

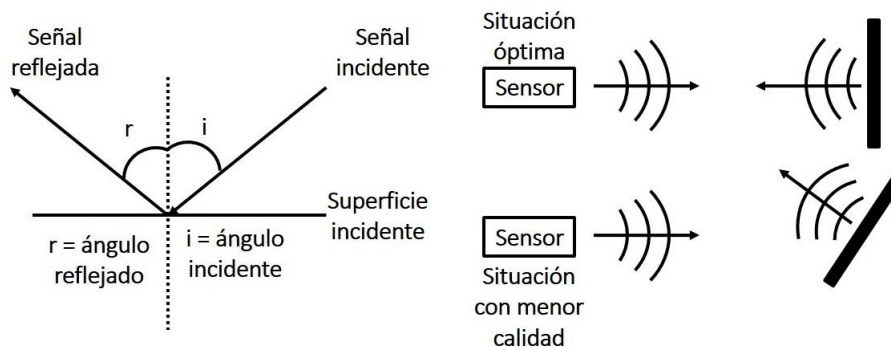


Figura 3.9: Explicación del error producido debido al ángulo de incidencia

- **Superficie.** Una superficie lisa agrava el problema del ángulo de incidencia. Cuanta más rugosidad, más superficie donde la onda podrá rebotar y por tanto, más probabilidad de que la lectura sea correcta. Se suele concluir que para que se produzca una buena reflexión, las irregularidades deben ser del orden de magnitud de la longitud de onda del ultrasonido.
- **Ambiente.** Las turbulencias debidas a las corrientes de aire pueden dificultar la detección del ultrasonido. La temperatura provoca cambios en la densidad del aire, y por tanto, en la velocidad con la que se propaga la onda, con el correspondiente error en la medición de la distancia.
- **Cercanía.** Algunos de estos sensores necesitan un intervalo de tiempo desde que envían la señal para preparar al receptor. Si el eco rebota antes de ese tiempo, no leerá la señal. Se suele decir que se necesita una distancia mínima para la detección de distancias debido a este tiempo mínimo de espera.
- **Rango de detección.** El campo de acción de la onda ultrasónica tiene una forma cónica, de forma que solamente se pueden obtener datos de los objetos dentro de dicho cono. Este hecho limita el rango de detección del sensor y además, supone una incertidumbre debido a que no se especifica la localización angular dentro de dicho cono.
- **Crosstalk.** Detección de falsos ecos, producida en aquellas áreas en las que se usan distintos sensores de ultrasonidos. Se produce *crosstalk* cuando una señal es recibida por un sensor distinto del emisor. También se producen errores cuando la onda sufre varias reflexiones antes de volver al receptor, lo cual proporciona una medida errónea de la distancia.

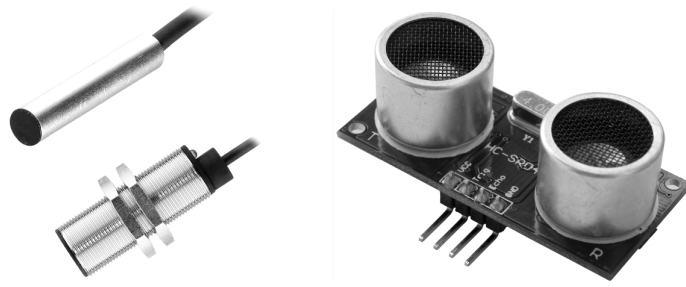


Figura 3.10: Sensores de efecto Hall (izquierda, Bernstein) y ultrasonidos (derecha, Arduino HCSR04 [9])

3.3.5. Sensores de Espectro Infrarrojo

Este tipo de sensores a menudo es definido en apartados de sensores de luz, pero su principal aplicación reside en el seguimiento o evitación de obstáculos, por lo que se ha decidido incluirlo en este apartado.

Los sensores infrarrojos utilizan parte del espectro denominado infrarrojo, invisible para el ser humano. A diferencia de otros sensores de luz, estos presentan una baja proporción de interferencias. No proporcionan información directa de la distancia hacia un objeto en general (existen modelos como GP2D02 y GP2D12 de la empresa Sharp que sí lo hacen), pero sí indican información acerca de si hay obstáculo o no en su cono de detección, que suele ser más estrecho que el de ultrasonidos.

Constan de una fuente luminosa (lámparas, diodos LED, láser, etc.) y una célula encargada de la recepción de la señal, que puede ser un fotodiodo o un fototransistor (ambos explicados en el apartado de sensores de luz). Suelen ser útiles para distancias de decenas de milímetros, o incluso algunos centímetros. El sensor de la Figura 3.11 de la empresa Sharp consigue medir hasta unos 24cm, lo cual no lo convierte en el más adecuado para su uso en UAVs, puesto que se trata de una distancia de muy corto alcance. Es necesario que los sistemas embarcados sean capaces de detectar objetos en largas distancias, para disponer del tiempo necesario para tomar una decisión y llevarla a cabo. Es posible que se use como sistema de visión de forma complementaria, como por ejemplo para actuaciones en el interior de edificios para rastreo de grietas en muros, pero siempre suelen estar integrados con algún otro de más alcance. Este es el fundamento del fenómeno “*Sense and Avoid*”, explicado más en detalle en el siguiente apartado de visión artificial.

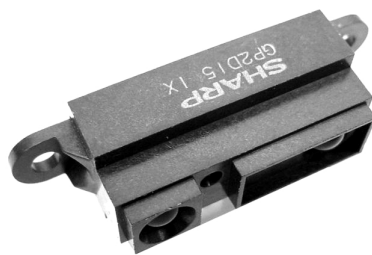


Figura 3.11: Sensores de infrarrojos, Sharp GP2D15)

Algunas de las aplicaciones donde su uso es común es en el seguimiento de un trazado de líneas, de paredes y detección de obstáculos en un circuito de baja velocidad. Tiene el inconveniente de ser sensible a la luz ambiente y a la reflectividad de los objetos.

3.3.6. Visión artificial

Cada vez más se pretende que los vehículos aéreos no tripulados sustituyan a los seres humanos en distintas labores de localización y reconocimiento, ya sea por medidas de seguridad, rapidez, eficacia o una combinación de todas ellas. Para que este fenómeno sea posible, es necesario dotar de una

visión tridimensional al UAV, puesto que el entorno en el que trabajarán tendrá dichas características espaciales. Aparte de esto, también será necesario implementar algún tipo de sistema que sea capaz de adaptarse a los cambios en sus alrededores, puesto que el entorno de trabajo real, fuera de laboratorio, es dinámico.

A su vez, el ámbito en el que se trabaja suele ser de tipo no estructurado, es decir, no se tienen conocimientos a priori y precisos de cómo se van a comportar los elementos que lo conforman. Por ejemplo, en el caso de reconocimiento de cultivos, se tienen conocimientos de lo que se espera ver, pero no se trata de una planta industrial en la que se conoce cómo van a actuar los robots manipuladores en cada momento.

Es por este motivo que es necesario dotar a los drones con capacidades sensoriales lo más similares posibles a las del ser humano dentro de lo posible. Para aclarar el concepto, [22] hace un símil entre los elementos del ojo biológico y el sistema visual artificial:

- Sistema de entrada de información equivalente al ojo: estaría formado por un conjunto de cámaras de vídeo y de algún sistema de adquisición de imágenes. Más adelante serán procesadas.
- Sistema de almacenamiento y procesamiento equivalente al cerebro: formado por una tarjeta de procesamiento y algún ordenador cuya CPU se encargue de realizar aquellos procesos que la tarjeta no esté capacitada.
- Sistema de salida/visualización (sin equivalente en el ojo biológico): suele ser un monitor para mostrar la información que ha sido procesada. Lo más recomendable es que sea en tiempo real para una mayor rapidez de actuación en caso de algún imprevisto.

Existe una amplia variedad de cámaras que pueden ser implementadas en UAV. Dependiendo de la función que vaya a realizar, de la capacidad de soportar carga del vehículo, de la resolución que se necesite, del precio que se esté dispuesto a pagar ..., se podrá escoger la más adecuada.

Existe un gran interés en desarrollar software y hardware capaces de proporcionar sistemas fiables dentro del ámbito del llamado “*Sense and Avoid*”. Este concepto hace referencia a la integración de todo tipo de sistemas capaces de realizar tareas similares a las que puede realizar un ser humano en cuanto a temas de seguridad y vigilancia. Para ello, es necesario el uso de sensores de visión artificial capaces de detectar obstáculos y calcular a qué distancia se encuentran para así evitar colisiones indeseadas. Se pretende mejorar la calidad del “*Sense and Avoid*” primero con vehículos aéreos no tripulados de pequeño tamaño, pero con el objetivo de en un futuro implementarlo en grandes aeronaves para reforzar el trabajo de los pilotos o incluso, en un futuro no se sabe cómo de lejano, sustituirlos. Existen dos funciones claramente diferenciadas:

- **Sense.** Hace referencia a la observación del llamado “intruso”, para obtener la mayor información posible acerca de sus características y régimen de vuelo. Como objetivos prioritarios se encuentran conocer el posicionamiento, rumbo y velocidad. En vuelos tripulados visuales VFR (del inglés, *Visual Flight Rules*), esta función se realiza a simple vista con ayuda de una simple radio. En vuelos instrumentales IFR (del inglés, *Instrumental Flight Rules*) o nocturno VFRN (del inglés, *Visual Flight Rules Night*) se precisan sistemas de ayuda a la visión del piloto, como son estos sensores de visión artificial.
- **Avoid.** Esta función es la encargada de analizar y procesar la información recibida del Sense. Debe tener la capacidad de decidir si el intruso detectado es conflictivo o no, si existe riesgo de colisión o no, y en caso afirmativo advertir al piloto. En el caso de los vehículos no tripulados, debe enviar información a la estación terrestre para alertar al piloto al mando. En el caso de los drones que se encuentren bajo el modo de piloto automático, esta función es autónoma, y por lo tanto, deberá determinar y ejecutar por sí misma la maniobra apropiada para evitar colisiones. Además, también es necesario que sea capaz de retornar a la ruta original establecida, por lo que el nivel de control automático implementado en estos sistemas es significativo.



Figura 3.12: Diferentes tipos de cámaras de visión artificial. U-Camera de Airelectronics, Hero 3 de GoPro, Tetracam ADC Light, Tau de FLIR (*Grupo Acre*)

En la Figura 3.12 se pueden observar cuatro cámaras. Las dos primeras proporcionan imágenes dentro del espectro visible tienen peso y tamaños reducidos y gran calidad de imagen. La tercera se trata de una cámara multispectral. Se utiliza sobre todo en cultivos, puesto que las imágenes infrarrojas (basadas en el principio explicado anteriormente) indican cambios en la vegetación antes de que aparezcan en el espectro visible. La última cámara se trata de una termográfica, basada en el principio de la radiación del cuerpo negro (en forma también infrarroja) en función de la temperatura. Los cuerpos con mayor temperatura emitan más radiación infrarroja que los que poseen menos temperatura. [12]

3.4. Sensores de luz

Estos sensores son capaces de cuantificar la presencia de luz usando una serie de dispositivos como pueden ser las células fotoeléctricas. Son utilizados para medir la intensidad de luz incidente, y algunos de ellos son capaces de orientarse para mejorar el aprovechamiento de los rayos solares. Una aplicación práctica que se le podría dar, sería la siguiente: en los drones que utilizan la energía solar como fuente de energía, instalar este tipo de sensores podría optimizar la orientación de las placas solares para sacarle el mayor partido a la incidencia de los rayos del sol.

Como se ha explicado anteriormente, dentro de este apartado se podrían incluir aquellas cámaras de vídeo con circuitería compleja, pero se decidió incluirla en el apartado de sensores de distancia y proximidad debido a su funcionalidad. Aquí se explicarán los sensores de luz sencillos, como pueden ser los fotodiodos, fototransistores y fotorresistencias.

1. **Fotorresistencias.** También llamado fotorresistor o LDR (del inglés, *Light-Dependent Resistor*), consiste en un dispositivo con una resistencia eléctrica cuyo valor varía en función de la luz incidente sobre él. Así, en los momentos en los que mucha (poca) luz incida sobre el sensor, esta resistencia disminuirá (aumentará). Esta variación suele producirse de manera no lineal, es relativamente lenta y además no se comporta de la misma forma al pasar de oscuro a claro que de claro a oscuro.

Ofrecen mayor sensibilidad a la luz que los fototransistores. Típicamente se fabrican con un cristal semiconductor fotosensible, como podría ser el sulfuro de cadmio (CdS), puesto que son sensibles a un amplio rango del espectro visible y no visible (infrarrojos y ultravioleta).

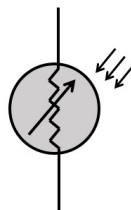


Figura 3.13: Esquema fotorresistencia

2. **Fotodiodos.** Este dispositivo de luz es comúnmente utilizado debido a que su respuesta a la luz es bastante rápida, de forma que se podría obtener un rango de tensiones lineal para un rango de

luminosidad. El fotodiodo está formado por un diodo semiconductor, construido como el diodo de unión-PN. En este caso, el semiconductor se encuentra expuesto a la luz debido a una lente transparente, y tendrá sensibilidad suficiente para la luz visible e infrarroja.

El fundamento de un diodo se basa en tener un sentido normal de circulación de la corriente, llamado polarización directa. En el sentido contrario no permite pasar corriente. Pero en este caso, el fotodiodo se encuentra polarizado de forma inversa, es decir, la corriente que se modifica con los cambios de intensidad lumínica, circula en sentido inverso al permitido por la juntura del diodo. Esta aumenta cuando el sensor es excitado por la luz.

Dependiendo de la aplicación se utilizarán unos fotodiodos u otros, cuyas diferencias residen en el material semiconductor utilizado (silicio, germanio, indio-galio-arsénico, sulfuro de plomo, etc.), con sus correspondientes rango de espectro.

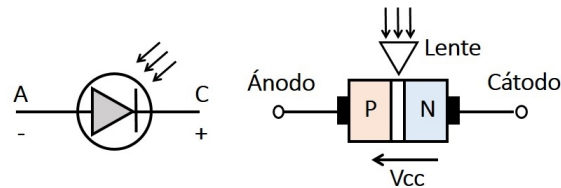


Figura 3.14: Esquema fotodiodo

3. **Fototransistores.** Al igual que el funcionamiento del fotodiodo se parecía al del diodo, este sensor funciona de manera similar a un transistor, con sus tres conexiones externas base-colector-emisor. De nuevo en este dispositivo se genera una corriente (colector-emisor) proporcional a la luz incidente en él (base-colector), gracias a una cápsula con una ventana transparente para el paso de la luz. De hecho, un transistor se puede convertir fácilmente en un sensor fototransistor conectando un fotodiodo entre colector y base.

Pueden proporcionar una corriente mucho mayor que las de un fotodiodo estándar, ya que son más sensibles a la luz que los anteriores. Tienen, al igual que los fotodiodos, un tiempo de respuesta muy corto. Una característica importante de estos fototransistores es que proporciona variaciones mayores de corriente como respuesta a variaciones de intensidad luminosa, debido a que cuentan con un factor de amplificación.

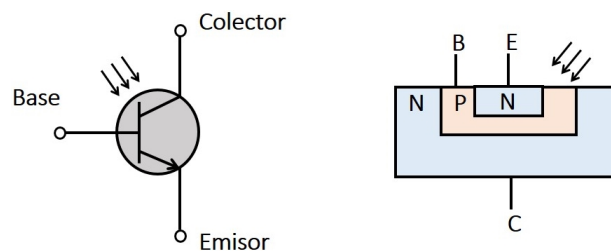


Figura 3.15: Esquema fototransistor

Aplicaciones inmediatas que un UAV puede extraer de este tipo de sensores sería por ejemplo en almacenamientos de naves industriales. El vehículo con un sensor de luz, podría detectar cuando una serie de objetos apilados alcanzan una altura determinada para dar la orden de comenzar a amontonar en otra columna diferente. También se podría utilizar para conocer y rellenar huecos vacíos entre una serie de objetos en una estantería, lo cual ahorraría el tiempo que emplearía un ser humano en detectar dicho espacio vacío.

3.5. Sensores de Velocidad

En el caso de los vehículos aéreos no tripulados, es necesario tener un control preciso sobre la velocidad a la que se está desplazando, para así poder mantenerla constante, anularla para dejarlo

estático o llevarla al máximo, por ejemplo. Se necesita que el sistema que acciona la velocidad de giro de los motores de sus hélices sea controlable desde un circuito remoto.

En este apartado se hablará de los **tacómetros** o **tacogeneradores**, los cuales son unos sensores capaces de medir velocidades angulares. Un tacogenerador sencillo se describe en la Figura 3.16, donde se observa que se utiliza un interruptor que utiliza fuerzas magnéticas para activarse o no. Esta activación dependerá de la cercanía o lejanía a la que se encuentre un diente magnetizado de la rueda dentada, de forma que cada vez que se acerque lo suficiente será accionado, y de esta forma se podrá obtener la velocidad de giro. Por cada vuelta, el dispositivo se activa y generará un voltaje de salida, el cual se utilizará posteriormente para calcular dicha velocidad.

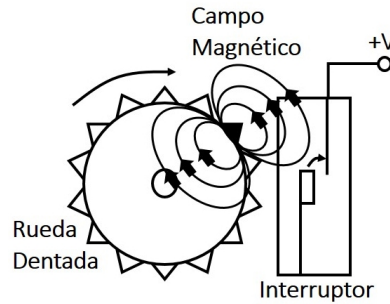


Figura 3.16: Tacogenerador: rueda dentada acciona el interruptor magnético [22]

El tipo más utilizado en la actualidad de sensor de velocidad es el tacómetro de corriente continua. Funcionan de manera inversa a los motores de corriente continua: a partir de una velocidad angular son capaces de obtener una tensión de salida proporcionales a la velocidad de giro, es decir, consiguen energía eléctrica a partir de mecánica. La Figura 3.17 explica el funcionamiento de este dispositivo. El rotor suele tener unas n bobinas, por lo que se producen $2n$ contactos al haber 2 estátor. Si el eje gira con una velocidad de ω , se generará una tensión alterna con un valor de

$$V_s = B_f S \sin(\omega t) \quad (3.5)$$

donde B_f es la inducción del campo magnético y S la superficie de la bobina en contacto. Después de calcular esta tensión alterna (proporcional a la velocidad de giro), se rectifica la tensión de manera que se obtenga una onda prácticamente constante, V_s en la Figura 3.17

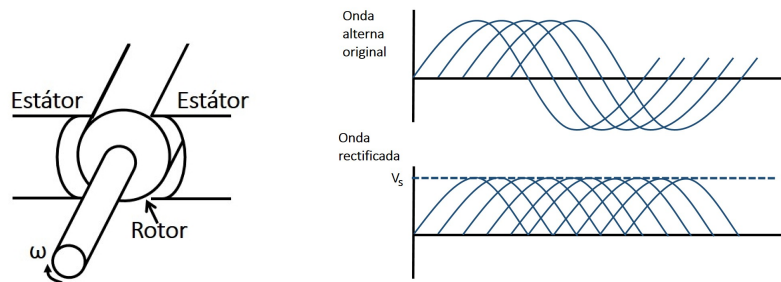


Figura 3.17: Esquema del funcionamiento de un tacómetro. [22]

La empresa SkyRC ofrece un modelo de tacómetro bastante apropiado para el aeromodelismo tanto de drones como helicópteros, el cual se puede observar en la Figura 3.18.



Figura 3.18: Tacómetro para UAVs y helicópteros de aeromodelismo, empresa SkyRC

3.5.1. Tubo de Pitot

También se puede conocer la velocidad que posee el dron midiendo la velocidad del aire que lo rodea. Para ello, se utiliza este sensor llamado tubo de pitot, el cual se encuentra representado en la Figura 3.19.

Consta de dos partes: una de ellas se encuentra expuesta a la presión total (dentro de la cavidad del tubo, abierta al flujo incidente) y la otra simplemente a la presión estática (presión ambiente, del fluido alrededor o atmósfera). La diferencia de presiones provoca que la membrana que se encuentra en el interior del sensor se desplace en un sentido u otro, una distancia proporcional a la diferencia de presiones dada. Esta distancia es medida y se puede obtener un voltaje de salida para procesar esa información de diferencia de presiones. La ecuación de Bernoulli es la que proporciona la velocidad según la diferencia de presiones

$$\frac{\rho V_a^2}{2} = P_t - P_e \quad (3.6)$$

donde ρ es la densidad del aire y V_a es la velocidad del UAV.

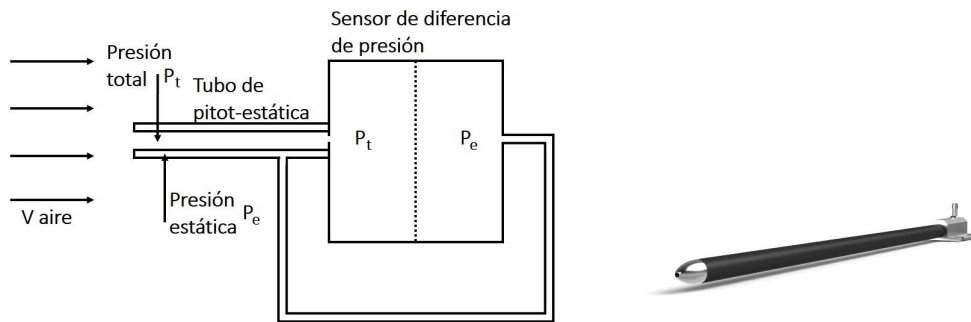


Figura 3.19: Esquema del tubo de pitot-estática con un sensor de presión junto con uno real para UAVs de la empresa UAV Factory [3]

Una fuente de errores común en este tipo de sensores es la presuposición de una atmósfera estándar, al igual que le ocurrirá a los altímetros. Puesto que las presiones y la densidad en la atmósfera varían debido a cambios meteorológicos, la atmósfera estándar no siempre proporcionará un modelo correcto para la determinación de la velocidad. Su calibración y reglaje se realiza en tierra para los UAVs (en aeropuertos para las aeronaves). También es importante la correcta colocación del tubo de pitot para que las tomas de aire se realicen tal y como se indica en las especificaciones del sensor. Una inclinación de este dispositivo puede suponer que la presión total tomada sea inferior a la real y por tanto, se produzca un error en la medida de la velocidad.

3.6. Codificadores (*Encoders*)

Los codificadores o también llamados *encoders* son dispositivos sensoriales electromecánicos capaces de transformar un desplazamiento en la posición angular o lineal en un código digital. Así, convierten una variación de un eje de rotación en una salida digital. Se basan en elementos que proporcionan información óptica para calcular el desplazamiento, cuyo funcionamiento se explica a continuación. Existen dos tipos de *encoders*, incremental y absoluto.

Este tipo de sensores es utilizado para calcular tanto posición como velocidad o aceleración. El principio en el que se basan es simple: la luz procedente de un LED o de una fuente de luz infrarroja pasa a través de un disco codificado de alta resolución que se encuentra rotando. Este disco contiene una serie de patrones, los cuales pueden encontrarse en binario, código Gray o BCD (*Binary-Code Decimal*). Un fotorreceptor que se encuentra al otro lado del disco, es capaz de captar los pulsos de luz que lo atraviesan según vaya rotando, y a su vez transforma esa información a forma digital como una cadena de pulsos binarios. Contando el número de pulsos generados se puede calcular la cantidad de grados girados.

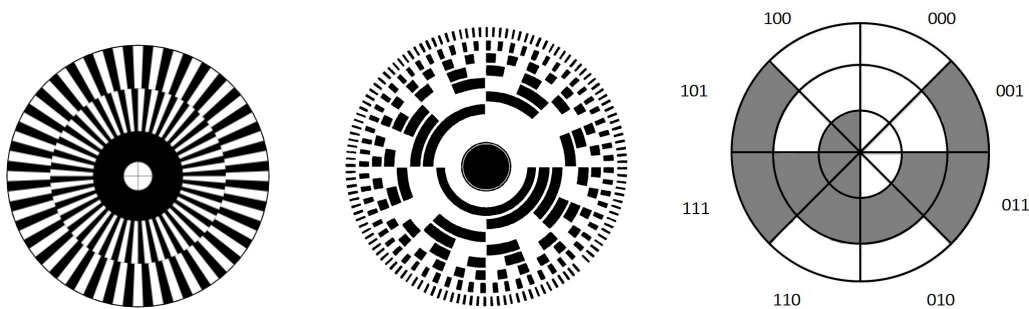


Figura 3.20: Ejemplos de encoders incremental, absoluto y código Gray

Esto significa que la resolución del codificador que se esté utilizando dependerá de la cantidad de marcas que tenga en el disco. En ocasiones se añaden marcas adicionales para conocer tanto el sentido de giro como el número de vueltas que lleva dado el disco. A continuación se explican los dos tipos de codificadores que existen, los incrementales y los absolutos.

- Encoder Incremental.** En este tipo de codificadores, la salida ofrece solamente una única descripción sobre la cantidad de grados que ha sido girado el eje. Si se precisa conocer la posición absoluta, sería necesario integrarlo con otros circuitos complementarios. Suelen ser más baratos debido a la baja precisión que poseen y son más sensibles a impactos.

Un inconveniente de este tipo de encoders sería que debe volver a la posición inicial una vez que se le deja de suministrar energía. En el caso de los absolutos no es necesario puesto que poseen una memoria no volátil.

- Encoder Absoluto.** Cumplen la misma función que el incremental solo que no es necesario implementar los circuitos complementarios para conocer la posición y dirección absolutas. Es muy corriente utilizar en este tipo de sensores un disco óptico codificado en código Gray. De esta forma, cada posición angular del fotorreceptor detectará una señal luminosa única. Los discos que lo componen poseen dibujos geométricos complejos, distribuidos en anillos concéntricos que representan los bits de la palabra binaria.

Por ejemplo, un disco que posea 8 anillos, tendrá una resolución de 8 bits, es decir, 256 porciones. Un disco con más anillos concéntricos ofrecerá mayor resolución de bits y dará posiciones angulares más precisas. Cubren un mayor rango de precio según la resolución necesaria.

3.7. Altímetros

Se utiliza este tipo de dispositivos para medir la altitud, es decir, la altura a la que se encuentra el vehículo con respecto del nivel del mar. Es importante para aquellas actuaciones de vuelo que requieran una altitud constante. Existen dos tipos de altímetros: los barométricos y los radioeléctricos.

- Altímetro barométrico.** Estos sensores miden la altura con respecto del nivel del mar basándose en la diferencia de presión que se produce por una diferencia de altura ($P_2 - P_1 = \rho g(z_2 - z_1)$). Su problema principal reside en que esta no es la única causa que genera variaciones en la presión, sino que además, los cambios en el clima (a “buen tiempo” altas presiones y a “mal tiempo” bajas presiones) y latitud (el nivel del mar no es uniforme), por ejemplo, también causan variaciones de presión. En este caso, sería necesario recalibrar el mismo altímetro si se utiliza en ambientes diferentes.

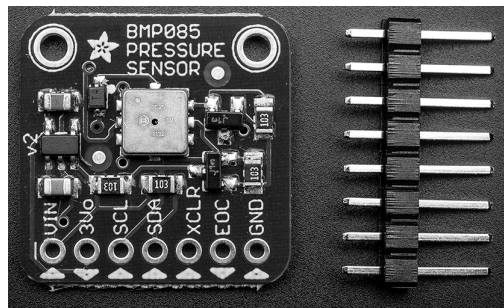


Figura 3.21: Altímetro barométrico BMP085 de Adafruits

- Radioaltímetros.** Los altímetros radioeléctricos no sufren los problemas de los barométricos. Su funcionamiento es bastante similar al de los radares: envían ondas de radio hacia la Tierra y esperan a que se refleje en ella. Puesto que se conoce a la velocidad a la que viajan las ondas (velocidad de la luz), se puede conocer la altitud a la que se encuentra respecto multiplicando esa velocidad por la mitad del tiempo transcurrido (la onda viaja de ida y vuelta). Son más rápidos, seguros y precisos que los barométricos, puesto que pueden detectar edificios, montañas, árboles y todo tipo de obstáculos que el UAV se encuentre a su paso.

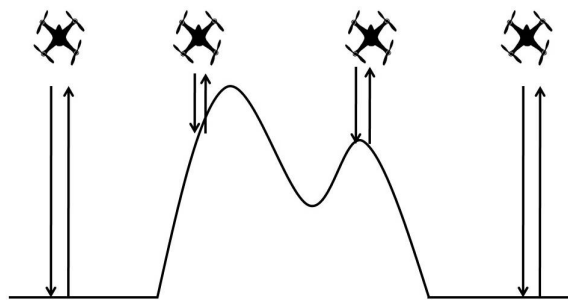


Figura 3.22: Esquema del funcionamiento de un altímetro radiométrico

3.8. Giroscopios, Acelerómetros y Magnetómetros: Los pilares de una IMU (del inglés, *Inertial Measurement Unit* - Unidad Inercial de Medida)

Para el ámbito de los UAV, donde prima el tamaño y peso de los componentes, son utilizados los llamados **giróscopos** o **giroscopios**, debido a su bajo coste y peso. Su funcionamiento se basa en el *principio de conservación del momento angular*, por el cual una rueda que se encuentra girando

se opone a un cambio del eje de rotación, o lo que es lo mismo, del plano de giro. Estos sensores proporcionan una gran estabilidad en los movimientos de vuelo.

Los giroscopios pueden medir la **velocidad de rotación** de una, dos o las tres direcciones del espacio. Si bien es cierto que estos sensores comenzaron utilizando inercia mecánica de los dispositivos que se encontraban girando, como los ejes de los rotores, actualmente están implementados con sistemas electrónicos y ópticos.



Figura 3.23: Esquema giroscopio y giroscopio MPU-6050 de 3 ejes compatible con Arduino, de empresa Oem

Para entender su funcionamiento se tiene la Figura 3.23. El disco simétrico se encuentra rotando alrededor de su eje de simetría (*Eje de rotación*). El dispositivo actúa de forma inesperada al aplicar una fuerza de rotación sobre el *Eje en el que se aplica la fuerza*, puesto que cambia su orientación no respecto a este eje, sino a un tercero (*Eje en el que se manifiesta la fuerza*) el cual es perpendicular a los dos anteriores. A este fenómeno se le conoce como efecto giroscópico, se encuentra visible en peonzas, pelotas de rugby, balas, etc., y es realmente útil para controlar la estabilidad de los sistemas que se basan en este principio.

A la derecha se puede observar un giroscopio real MPU-6050 compatible con Arduino. En el ámbito de los UAVs, es normal el uso complementario de giroscopios con acelerómetros, de tres ejes cada uno, y así tener cubiertos los 6 grados de libertad del vehículo.

Los **acelerómetros** son sensores que sirven para conocer la orientación con respecto a unos ejes de referencia que posee un dispositivo, por lo que tiene una gran utilidad para el tema de control y estabilidad de los drones. Se basan en la aceleración constante producida por la fuerza de la gravedad 9.81 m/s^2 , mediante la cual son capaces de medir aceleraciones en los ejes según estén orientados, teniendo como referencia que la aceleración de la gravedad es perpendicular al suelo.

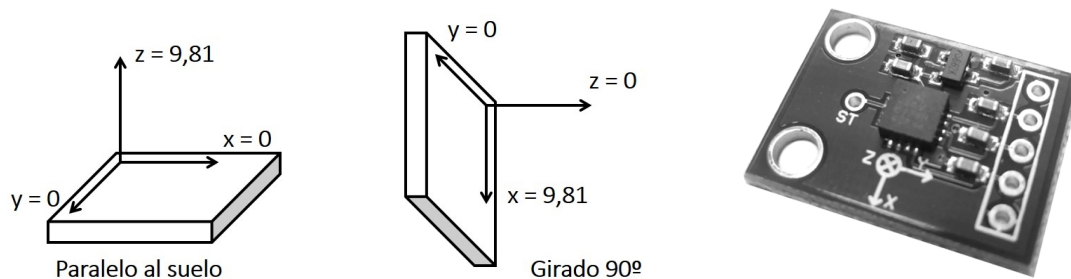


Figura 3.24: Medidas de acelerómetro y acelerómetro Arduino Adxl335 de 3 ejes

En la Figura 3.24 se explica su funcionamiento. Si se encuentra paralelo al suelo, medirá una aceleración en el eje z de 9.81 m/s^2 . En el caso de que se gire el dispositivo, medirá una nueva aceleración en otro eje (en este caso se mide la aceleración completa en el eje x por haber girado 90°), y así, se podrá calcular su orientación.

Los **magnetómetros** son dispositivos que miden campos magnéticos a partir de la densidad de

flujo magnético que les atraviesa. Puesto que la Tierra genera un campo magnético, los magnetómetros son capaces de actuar como brújulas para conocer en todo momento dónde se encuentra el norte magnético terrestre. Son utilizados en los drones para conocer en cada momento su orientación y la dirección de su movimiento.

Un problema que poseen este tipo de sensores viene cuando se encuentran cercanos a otros dispositivos electrónicos o a cables que conducen corriente. En estos, puesto que conducen una corriente, el campo magnético que ésta genera puede causar variaciones en las medidas del magnetómetro. Es por ello que se recomienda su uso en una zona que se encuentre elevada o alejada de cualquier otro objeto metálico dentro del UAV.



Figura 3.25: Magnetómetro de 3 ejes HMC5883L, de Naylamp Mechatronics

Actualmente, tanto giróscopos como acelerómetros y magnetómetros se pueden construir con un solo chip de silicio, incluyendo en él la parte electrónica encargada de manejar y procesar las señales. Este dispositivo que los integra se conoce como **IMU** (del inglés, *Inertial Measurement Unit*). Estos sistemas son utilizados para conocer en todo momento la orientación del dispositivo y la trayectoria que está siguiendo. Son realmente útiles cuando la señal GPS no está disponible o tiene poca calidad, esto es, cuando el vehículo se encuentra en zonas donde no existe contacto entre el emisor y el receptor GPS, como pueden ser túneles, dentro de edificios, o existen interferencias debidas a componentes electrónicos.

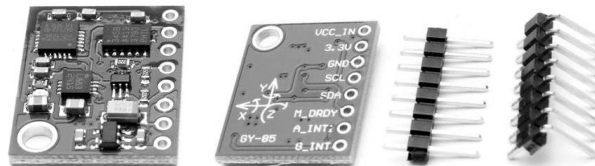


Figura 3.26: Módulo de Sensor IMU GY-85 de 9 ejes (Giróscopo, Acelerómetro y Magnetómetro), de Banggood

Como se puede observar en las fotografías de ambos, son dispositivos con un tamaño realmente reducido, gracias al uso de tecnología MEMS (del inglés, *Microelectromechanical Systems*). Además, son bastante versátiles y su costo es accesible.

Actualmente, los drones utilizados a cualquier nivel, comercial o particular, suelen estar equipados con una IMU como mínimo (a veces junto a un receptor GPS), la cual incluye normalmente un giróscopo, acelerómetro y magnetómetro de 3 ejes cada uno. Estos sensores son susceptibles a poseer errores debido a varias causas: daños en los sistemas, variaciones de temperatura o vibración excesiva, entre otras.

Son numerosos los estudios de investigadores en los cuales se pretende solventar el problema de los errores producidos en la IMU. Se basan principalmente en una linealización del modelo dinámico del vehículo alrededor de un punto de equilibrio. Sin embargo, la dinámica de un UAV es altamente no lineal, y las medidas sobre el estado del sistema completo no se encuentran siempre disponibles para muchas aplicaciones prácticas. [2]

Estos dispositivos además proporcionan una medida fiable sobre la velocidad transversal del vehículo no tripulado. Dicha variable es crucial para obtener una navegación completamente autónoma, puesto

que para numerosas aplicaciones será necesario controlar un vuelo rectilíneo. La IMU proporciona unos datos sobre orientación capaces de ser procesados y utilizados para realizar dicha acción, y por ello su uso es tan extendido en este ámbito de los drones.

En algunos estudios [15], se utilizan IMUs junto con cámaras para calcular una estimación de la velocidad. Como viene siendo normal en numerosos sistemas, se integran los datos de ambos sensores para obtener una medida más fiable, y en este caso se utiliza un Filtro de Kalman, al igual que en este trabajo. En este artículo se utilizan las imágenes de la cámara para conocer las coordenadas del punto donde se encuentran combinadas con las lecturas del acelerómetro.

La principal desventaja que se les puede encontrar a estos sensores es la acumulación de errores con la que trabajan. Este dispositivo calcula su nueva posición basándose en las posiciones previamente calculadas. Por lo tanto, cualquier error que se produzca en la medida, por muy pequeño que sea, se va acumulando de punto a punto. Este fenómeno provoca que a medida que pasa más tiempo desde que se tomó la medida inicial, la posición real difiera más y más de la posición calculada.

La tasa de muestreo también es un factor que se encuentra limitado por los giróscopos y acelerómetros que integran la IMU. Esto es debido a que los dispositivos son capaces de tomar datos en un intervalo de tiempo finito. Por ejemplo, si el acelerómetro es capaz de tomar medidas una vez cada segundo, la IMU interpretará que esa aceleración se ha producido durante todo el segundo, incluso cuando puede que ocurran grandes variaciones en ese intervalo de tiempo. La resolución del sensor dependerá de la calidad de los componentes que la integren, esto es, de los acelerómetros y giróscopos. Cabe destacar que un error constante en la aceleración se traduce en un error lineal en la velocidad y en un error cuadrático en la posición, por lo tanto, es recomendable una correcta calibración y tener cuidado con el uso de estos dispositivos.

3.9. GPS - Global Positioning System

El sistema de posicionamiento global GPS es un sistema de navegación de gran precisión y de operación continua basado en el uso de satélites que proporcionan información sobre la posición en 3D para objetos sobre la superficie terrestre. Fue desarrollado por el Departamento de Defensa de los Estados Unidos, y ha estado completamente operativo a partir del año 1993. Para los vehículos aéreos no tripulados, es difícil cuantificar la importancia del desarrollo de este sistema de navegación. Fue y sigue siendo una **tecnología crítica** en el montaje de pequeños UAVs. En esta sección se describirá brevemente su funcionamiento y por qué es un “*must*” de los drones cuando no todos los sensores presentados con anterioridad tienen esta característica.

El sistema GPS consta de una constelación de 24 satélites orbitando alrededor de la Tierra a una altura de 20.180 km. Se encuentran dispuestos de tal manera que cualquier punto de la superficie es observado siempre por 4 de esos satélites como mínimo. Estos satélites envían señales a un receptor que se encuentre cercano a la superficie terrestre, y se calcula su posición basándose en el tiempo que tarda dicha señal en recorrer el trayecto. Puesto que existen errores de sincronización en los relojes tanto de los satélites como de los receptores (a pesar de ser atómicos y tener gran precisión), a la distancia calculada por el tiempo de vuelo de la señal se le denomina *pseudorange*, para diferenciarlo del verdadero rango (corrigiendo esos errores de sincronización).

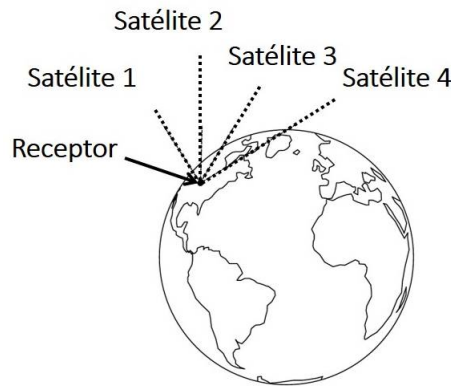


Figura 3.27: Medida de los pseudorangos de 4 satélites usados para triangular la posición del receptor [3]

En la Figura 3.27 se observan los 4 satélites necesarios para triangular la posición en 3D del receptor, en este caso un UAV. ¿Por qué son necesarias cuatro medidas del pseudorango? Pues bien, para calcular una posición en tres dimensiones bastaría con utilizar tres medidas diferentes, así se obtendrían las coordenadas (x,y,z) o $(\text{latitud}, \text{longitud}, \text{altitud})$. Pero para resolver el problema del offset de los relojes, se necesita una medida adicional. De esta forma se obtendría un sistema de cuatro ecuaciones no lineales con cuatro incógnitas: latitud, longitud, altitud y el offset del reloj del receptor.

El GPS es el más desarrollado y usado de los diferentes sistemas GNSS (del inglés, *Global Navigation Satellite System*). En Febrero del 2012, el Congreso de los Estados Unidos instó a la Agencia Federal de Aviación **FAA**, (del inglés *Federal Aviation Agency*), que llevara a cabo un plan para la seguridad de la rápida incorporación de los UAVs dentro del ámbito civil. Se pedía un aumento de seguridad para la navegación de este tipo de vehículos puesto que utilizaban la señal de GPS de uso comercial y no la de uso militar, la cual tiene una mayor precisión y seguridad. Como ejemplo, la Universidad de Texas demostró que era capaz de acceder a la señal GPS de uso civil y modificarla, introduciendo información falsa en dichas señales (fueron capaces de desviar el curso de un velero cientos de metros que se encontraba en el Mediterráneo). Una medida para no estar completamente expuestos a estos ataques de intrusismo, tal y como se hace actualmente y se propone en [19], es utilizar fuentes independientes que proporcionen información sobre la navegación. Por ejemplo, el avión comercial B-787 actualmente utiliza varios sistemas de GNSS como GPS (EEUU), BeiDou (China), GLONASS (Rusia) y Galileo (Europa). Con varias señales que proporcionan la misma información es más complejo acceder y modificar los datos.

Al igual que en todos los sistemas sensoriales, el GPS no se libra de poseer fuentes de errores que son necesarias tener en cuenta para mitigar sus efectos. La precisión de la medida que se obtiene de la posición dependerá tanto de la precisión de los *pseudorangos* como de la geometría de los satélites que el receptor tiene a su alcance. La mejor posición es formando un tetraedro con el receptor en el centro de una de las caras, y el vértice opuesto a dicha cara sobre el usuario. Para tener en cuenta los errores debidos a la geometría, se utilizan una serie de factores denominados **DOP** (del inglés, *Dilution of Precision*). También se producen errores debido al tiempo de propagación de la señal. Puesto que estas viajan a la velocidad de la luz, porque son señales electromagnéticas, un error de 10ns podría resultar en un error en posición de hasta 3m. Algunas fuentes que provocan errores en los tiempos de propagación son brevemente descritos a continuación.

- **Datos de la efemérides.** La efemérides es una descripción matemática de la órbita del satélite. Para calcular la posición del receptor es necesario conocer la del emisor con precisión, es decir, no cometer errores en esa descripción matemática sobre su órbita.
- **Reloj del satélite.** Los relojes utilizados son atómicos, fabricados de cesio y rubidio. A lo largo del día arrastran un error de 10ns, por lo que como se dijo anteriormente provocan alrededor de 3m de error en posición. Este problema se soluciona actualizando los relojes cada 12 horas, por lo que el error en posición medio es de 1 o 2 metros.

- **Ionosfera y Troposfera.** La ionosfera es la capa más externa de la atmósfera, donde se producen los mayores errores en la propagación (alrededor de 2 y 5 metros) debido a la presencia de electrones libres. La troposfera, por el contrario, es la capa más baja de la atmósfera, donde se concentra la mayor masa de la atmósfera. Ahí se produce la mayor actividad meteorológica, por lo que los cambios de presión, temperatura y humedad afecta a la velocidad de la señal enviada. Los errores que ésta introduce son cercanos al metro.
- **Multitrayectoria.** Se produce cuando el receptor recibe señales reflejadas en edificios u otros grandes obstáculos que provocan errores por debajo de un metro en la mayoría de las circunstancias. Estas señales reflejadas ocultan y dañan la medida real de la posición.

La cantidad de información que se obtiene a partir del uso de GPS es bastante llamativa. En el propio código de la señal que llega al receptor se indican valores no solo de posición, sino también de velocidad, número de satélites utilizados, hora UTC (del inglés, *Coordinated Universal Time*), DOPs, elevación y azimut de satélites, calidad del mensaje, etc. Todos estos apartados se verán con más profundidad a la hora de analizar los datos obtenidos en el formato NMEA proporcionado por el receptor GPS.

3.9.1. Segmentos

Este sistema de posicionamiento y navegación consiste en tres segmentos: segmento espacial, segmento de control y segmento de usuario. Las Fuerzas Aéreas de los Estados Unidos son las encargadas del desarrollo, mantenimiento y operación de las dos primeras.

- **Segmento Espacial.** Se trata de la constelación de los satélites transmitiendo radio señales a los usuarios. Consta de 24 satélites operacionales el 95 % del tiempo. Se encuentran distribuidos en 6 planos orbitales, con 4 satélites situados en cada plano. Esta disposición permite que cada usuario pueda ver al menos cuatro satélites desde cualquier punto del planeta.

Vuelan en órbitas medias denominadas **MEO** (del inglés, *Medium Earth Orbit*), las cuales son circulares, y a una altura de 20.200km aproximadamente. Cada satélite orbita alrededor de la Tierra dos veces al día.

- **Segmento de Control.** Consiste en la red de instalaciones que monitorizan los satélites de GPS, interpretan las transmisiones, realizan análisis y mandan instrucciones y datos a las constelaciones. Son las encargadas de actualizar la posición real de los satélites (las efemérides) basándose en observaciones, además de sincronizar los relojes atómicos. Las estaciones de control se encuentran distribuidas por todo el planeta, y consta de: una estación de control principal (en Colorado, EEUU), una estación de control principal alternativa (en California, EEUU), 11 antenas de control y orden, y 15 lugares de monitorización.

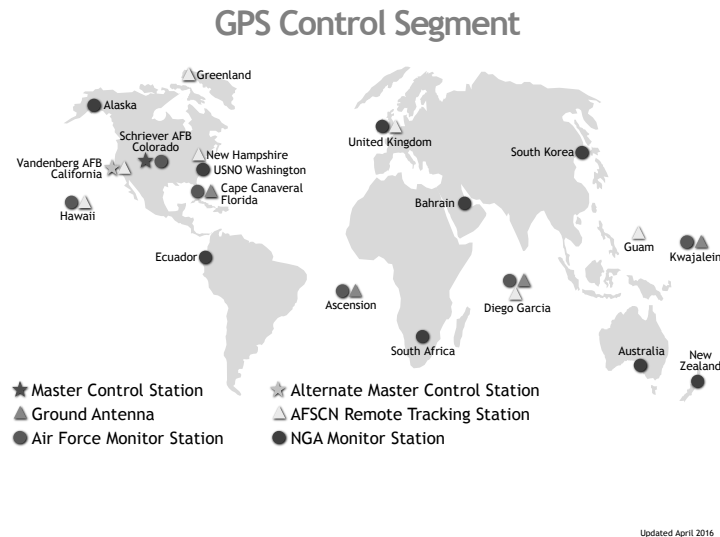


Figura 3.28: Segmentos de Control de GPS

- **Segmento de Usuario.** El GPS se ha convertido en los últimos años en uno de los pilares tecnológicos en los que se basan numerosas aplicaciones que se utilizan de forma cotidiana. El segmento de usuario consta de los receptores que poseen los usuarios de GPS para obtener su posición a partir de las señales recibidas. Este receptor debe contener un reloj de cuarzo para sincronizarlo con los relojes de los satélites y así obtener una medida más fiable de la posición. Estos receptores deben también decodificar el mensaje enviado por el satélite. Suelen estar formados por: antenas, receptores, microprocesadores, almacenamiento de datos para el cálculo de rutas, unidades de control para facilitar el uso al usuario, entre otras. Algunas de las prestaciones de este dispositivo son las siguientes:

- Actualización de la posición de 0.5 a 1 segundo
- Precisión en torno a los 15m
- Precisión de 0.1m/s en cuanto a la velocidad del usuario
- Precisión de 100ns en cuanto a la referencia temporal

3.9.2. Disponibilidad, Integridad y Continuidad

La principal limitación con la que cuenta este sistema es con la dependencia de los Estados Unidos, más en concreto con el departamento de defensa. Se están desarrollando otros sistemas de posicionamiento global como son el Galileo (Europa) o GLONASS (Rusia) para paliar esta dependencia, ya que en cualquier momento podrían decidir limitar el uso del GPS por parte de la población civil.

Actualmente el sistema GPS es el que consigue una mayor precisión con respecto al resto de sistemas de navegación en uso. Sin embargo, la precisión de un sistema no es la única variable a tener en cuenta a la hora de utilizar un sistema u otro, puesto que existen otros conceptos como pueden ser su disponibilidad, integridad y continuidad. En este apartado se describen brevemente.

- **Disponibilidad.** Porcentaje de tiempo que dicho sistema se encuentra utilizable, dentro de su área especificada de cobertura. El concepto de utilizable hace referencia a que cumpla unos requisitos mínimos previamente especificados para que la información que proporciona sea mínimamente fiable. Un ejemplo típico sería limitar alguno de los DOP a valor concreto, por debajo del cual no se tendrá en cuenta la señal recibida y por tanto los datos contenidos en ella. El área de cobertura del GPS incluye toda la superficie terrestre, aunque hay que tener en cuenta el ángulo de elevación del satélite en el horizonte a partir del cual se considera visible para el receptor, llamado “*ángulo de máscara*”. Éste será mayor en entornos urbanos que despejados.

- Integridad.** Capacidad del sistema para controlar que el propio sistema no debe ser utilizado, ya sea porque no se encuentre operativo o porque posea errores. Se le considera un parámetro que equivaldría a una medida de confianza que se pueda tener en el sistema. Los efectos de radiación del espacio que alteren los relojes o la electrónica, los fallos de los satélites, o el error humano de software o hardware en el segmento de control son algunas de las causas que provocan que la integridad del sistema no se la esperada. La probabilidad de que sucedan este tipo de fallos son bajos, pero inadmisibles para aplicaciones de navegación aérea.

Puesto que se trata de un parámetro realmente importante puesto que en él reside la confianza en el sistema, se han desarrollado diversos procedimientos para aumentar la integridad del GPS. Una de las técnicas más conocidas es el **GPS diferencial** o **DGPS** (del inglés, *Differential GPS*). Se trata de utilizar una serie de estaciones en tierra (**GBAS**, del inglés, *Ground-Based Augmentation Systems*) cuya posición se conoce con gran precisión. Estas están equipadas con un receptor GPS y se encuentran en comunicación con el usuario, para proporcionar mayor precisión a la señal del satélite.

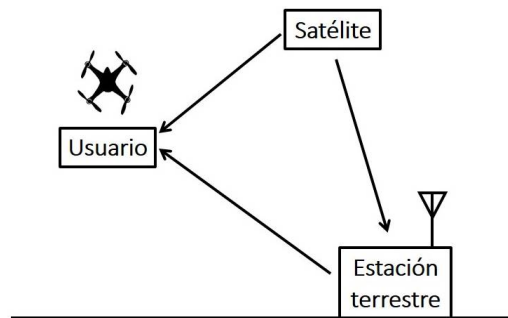


Figura 3.29: Ejemplo de DGPS basado en GBAS

- Continuidad.** Probabilidad de que el sistema de navegación pueda ser usado de manera continua durante la realización de una misión u operación. Que pueda ser usado hace de nuevo referencia al concepto previamente definido de “utilizable”, es decir, que cumpla unos requisitos mínimos impuesto por la operación a realizar, y que de nuevo podrían ser expresados en términos de los DOPs. Dependiendo de la misión u operación que se lleve a cabo, la continuidad será una u otra, pero en cualquier caso se encontrará íntimamente ligada con fallos no planificados en los satélites. La probabilidad de que en un satélite se produzca un fallo que produzca que dicho satélite deje de emitir de forma no planificada es del 0.0001 %.

Muchos sistemas militares de navegación inercial podrían ser reemplazados con sistemas con menos fiabilidad y coste si se garantizara la continua disponibilidad del GPS. Esta continua disponibilidad serviría para actualizar al sistema inercial y limitar así su error, el cual se propaga a lo largo del tiempo. Puesto que no está garantizada, se desarrollan sistemas alternativos de navegación inercial más costosos pero cuya fiabilidad es mucho mayor, debido a que se conoce que no va a sufrir ataques externos. [19]

4 | Filtro de Kalman

Esta sección se centrará en la explicación, formulación y análisis de una herramienta muy comúnmente utilizada, el Filtro de Kalman. Será necesario comprender sus expresiones y algoritmos para el problema que se encuentra bajo estudio. Es muy típico encontrar este tipo de procedimiento en multitud de sistemas, ya que ofrece una amplia gama de aplicaciones en diversos ámbitos.

En este caso, puesto que el objetivo final es implementar en Matlab una simulación mediante la cual se puedan integrar dos señales recibidas por un UAV, señal GPS y señal de una IMU. Se explicarán también otros tipos de filtros utilizados actualmente.

(EXTENDER CUANDO TERMINE)

4.1. Introducción

El filtro de Kalman fue desarrollado en primer lugar para su uso en la navegación de aeronaves, pero su implementación se ha visto ampliada, abarcando todo tipo de ámbitos y campos. Su principal uso reside en estimar los estados de un sistema, de los cuales se ha hablado con anterioridad en este trabajo. Dichos estados suelen tener una propiedad común la cual hace necesaria que se le apliquen este tipo de filtros: solamente pueden ser observados de forma **imprecisa** por el propio sistema. Esto es debido a que el estado de un sistema es proporcionado por los datos procedentes de los sistemas sensoriales anteriormente detallados, los cuales se han visto que poseen diversas **fuentes de ruido que producen errores** en las medidas. Por este motivo se ha decidido aplicar este tipo de sistema de filtrado para mejorar la calidad de las señales recibidas (en el caso de GPS) o de los datos obtenidos (en el caso de la IMU). Existen otro tipo de razones por las cuales ha sido elegido este filtro, como su atractivo debido a que minimiza la varianza del error de la estimación, las cuales serán explicadas más adelante.

El objetivo de este apartado es introducir los conceptos básicos sobre los algoritmos del Filtro de Kalman, y utilizarlos en una **simulación** para resolver un problema real sobre la navegación de un vehículo, en este caso aéreo y no tripulado. Este método proporciona una herramienta suficientemente fiable para estimar la posición (o cualquier otra variable a controlar) actual del vehículo.

En el momento de la obtención de datos mediante las señales mencionadas anteriormente, es inevitable que se produzcan efectos indeseados de ruido, los cuales pueden corromper el mensaje. Un buen algoritmo de filtrado debería ser capaz de eliminar este ruido de las señales (el acumulado al utilizar un acelerómetro, al atravesar la atmósfera la señal de GPS, etc.) a la vez que retiene la información útil que se desea obtener. Es por ello que anteriormente se han detallado las fuentes de errores de todo tipo de sensores, porque un mejor conocimiento del origen de este error proporciona información necesaria para diseñar un modelo lo más realista posible.

4.2. ¿Qué es el filtro de Kalman?

El filtro de Kalman es un proceso matemático iterativo que utiliza una serie de ecuaciones y datos de entrada constantes para rápidamente estimar el valor real de la posición, velocidad, o cualquiera que sea la variable, del objeto medido, cuando los valores medidos contienen errores impredecibles o aleatorios, incertidumbre o variación. Se dice que es un proceso rápido puesto que es capaz de acercarse al valor real tomando un número relativamente bajo de esos datos de entrada.

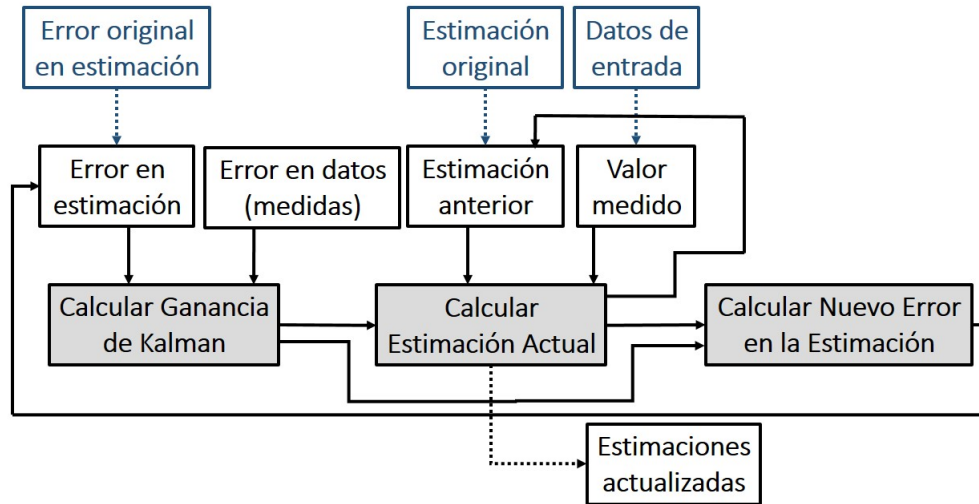


Figura 4.1: Proceso iterativo del Filtro de Kalman

En la Figura 4.1, obtenida de [24], se muestra a grandes rasgos, un diagrama de los principios en los que se basa en filtro de Kalman. Será realmente útil cuando llegue el momento de explicar las tres principales ecuaciones o cálculos necesarios. En la misma figura se puede ver que de hecho se trata de un proceso iterativo.

El primer término que se necesita calcular, en cada iteración, es el llamado “*Ganancia de Kalman*”. Después, será calculado la “*Estimación Actual*”, la cual significa que se procederá a actualizar el valor estimado. Por último, el “*Nuevo Error en la Estimación*” será recalculado a partir de los dos cálculos anteriores. Se utilizará por comodidad el término error siempre como sinónimo de incertidumbre. Los cálculos de estos tres términos determinarán las tres principales ecuaciones mencionadas anteriormente.

Estos son los requisitos y la funcionalidad de cada uno de los términos:

- ***Ganancia de Kalman***. Se necesitarán dos datos, el error en la estimación y el error en la medida, como se puede observar en el diagrama de la Figura 4.1 (dos flechas llegan a la ganancia de Kalman).
 - Primero, el **error en la estimación**, que puede ser tanto el error en el anterior o si es el caso de la primera iteración, el error original. La estimación, como se verá más adelante, está basada en el modelo dinámico que se emplee para modelar el sistema. Por ejemplo, en el caso del seguimiento de una trayectoria de un UAV, el modelo se regirá por las leyes de Newton de la Cinemática $x = x_0 + vt + 1/2at^2$. El error en la estimación se produce en el intento de pasar a modelo matemático un problema real, puesto que se cometen fallos tanto debido a aproximaciones, como a redondeos, simplificaciones, etc.
 - Segundo, el **error en el dato de entrada** recibido en ese instante. Éste es debido a los errores que se cometen al tomar medidas con sensores, y dependerá del sistema sensorial que se esté utilizando. En el apartado anterior se han explicado los diferentes errores que se cometen con los diferentes sensores. En el caso bajo estudio, donde se tomarán datos de un GPS y una IMU, los errores serán los correspondientes a dichos sistemas. Por comentar algún ejemplo, el GPS comete errores de sincronización entre sus relojes, y la IMU acumula errores a lo largo de la trayectoria.

El objetivo de la *Ganancia de Kalman* será ponderar una importancia relativa en el primer error (de la estimación E_{EST}) versus el segundo error (dato de entrada), llamado en este caso E_{MEA}

debido al inglés *measurement*.

$$KG = \frac{E_{EST}}{E_{EST} + E_{MEA}} \quad KG \in [0, 1] \quad (4.1)$$

Se ha querido explicar el significado de esta ecuación mediante el esquema de la Figura 4.2. La *Ganancia de Kalman* puede tomar valores comprendidos entre 0 (baja) y 1 (alta). ¿Qué significa obtener una ganancia de Kalman alta? Como se observa en el esquema, una ganancia de Kalman alta implica que el Error de la Estimación es **grande** comparado con el Error de la Medida. Y si el Error de la Medida es más pequeño que el Error que se ha estimado, **interesa** tener más en cuenta la medida (dato obtenido), y “despreciar” el valor de la estimación porque contiene un error alto. Este concepto de dar más importancia a un valor que a otro es la función principal de la ganancia de Kalman, como se verá al realizar el siguiente cálculo: *Estimación Actual*.

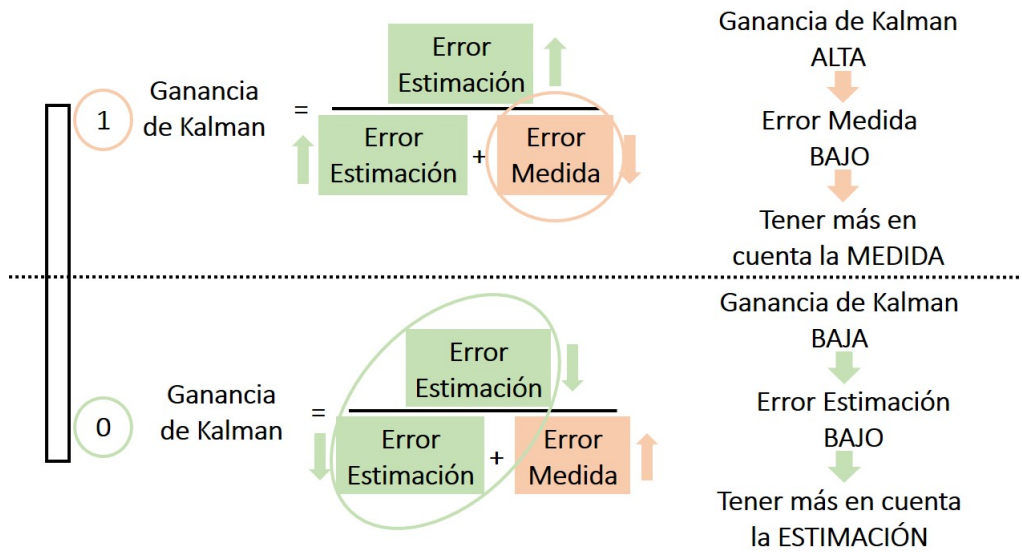


Figura 4.2: Explicación del uso de la Ganancia de Kalman

- **Estimación Actual.** Para el cálculo de la estimación actual EST_t se necesitarán tres variables (tres flechas llegan a la estimación actual), tal y como se puede ver en el diagrama de la Figura 4.1: la estimación anterior EST_{t-1} , el valor medido por el sensor MEA , y la ganancia de Kalman calculada anteriormente KG . Esta estimación se rige por la siguiente expresión, en las cuales aparecen dichas variables

$$\begin{aligned} EST_t &= EST_{t-1} + KG[MEA - EST_{t-1}] \\ &= EST_{t-1}[1 - KG] + KG * MEA \end{aligned} \quad (4.2)$$

- Primero, la **estimación anterior**. Como se ha dicho con anterioridad, el filtro de Kalman es un proceso iterativo, y por lo tanto, en cada iteración, cualquiera que sea la estimación actual para un tiempo t pasará a ser la estimación anterior en el tiempo $t+1$. En el diagrama de la Figura 4.1 se observa que la estimación anterior se puede obtener de dos formas: si es la primera iteración se le asignará un valor original que se considere oportuno, y si se trata de la segunda o siguientes iteraciones, corresponderá a la anterior estimación actual. Una de las características más importantes del filtro de Kalman que lo hacen tan especial reside en dicho “valor original” que se le asigna al comienzo del proceso. Este filtro tiene la propiedad de que no importa qué valor es determinado como inicial, puesto que rápidamente será capaz de acercarse al valor real basándose en los errores.
- Segundo, el **valor medido por el sensor**. Este es el dato de entrada que se recibe de los sensores, en este caso, del GPS y/o de la IMU. Estos sensores proporcionarán datos sobre

posición y velocidad que, como ya se ha explicado, contienen fuentes de **errores**, y según el grado de error que contengan las señales, se tendrán más o menos en cuenta a la hora de calcular la nueva estimación.

- Y tercero, la **ganancia de Kalman**. Pues bien, ha llegado el momento de decidir. En este punto se tiene: una estimación anterior (a partir de la cual, con el modelo matemático correspondiente se conseguirá calcular la estimación actual) con su correspondiente error, y unos nuevos datos de entrada con sus correspondientes errores. ¿Cómo decidir a qué valor de estos darle más importancia? ¿Se toma uno y se desprecia el otro por completo? Para responder a estas preguntas se dispone de la ganancia de Kalman y de la ecuación 4.2. La ganancia de Kalman será la herramienta utilizada para ponderar entre ambos valores basándose en cuál de los dos posee un mayor error, como se indicó en la Figura 4.2. Atendiendo a la ecuación, se observa cómo,
 - si la ganancia de Kalman es alta ($\mathbf{KG}=\mathbf{1}$, error en la estimación alto), el término $EST_{t-1}[1 - KG]$ se anularía puesto que contiene alto error, mientras que el término $KG * MEA$ se mantendría porque el error en la medida es bajo (en comparación con el de la estimación).
 - si la ganancia de Kalman es baja ($\mathbf{KG}=\mathbf{0}$, error en la medida alto), el término $KG * MEA$ se anularía porque contiene alto error, mientras que el término $EST_{t-1}[1 - KG]$ se mantendría puesto que el error en la estimación es bajo (en comparación con el de la medida).
- **Nuevo Error en la Estimación.** Para este apartado se necesitarán dos variables (dos flechas llegan a este punto en la Figura 4.1): la estimación actual y la ganancia de Kalman. Es decir, los dos pasos anteriores son necesarios para calcular este error.

$$E_{EST_t} = \frac{E_{MEA}E_{EST_{t-1}}}{E_{MEA} + E_{EST_{t-1}}} \implies E_{EST_t} = [1 - KG]E_{EST_{t-1}} \quad (4.3)$$

Es en este punto en el que comienza la nueva iteración, y por lo tanto el término $E_{EST_{t-1}}$, hace referencia al error en la estimación calculada justo en el apartado anterior, en donde se hablaba de actual pero que ya ha pasado a ser la anterior.

Es interesante destacar que el error en la estimación actual siempre será más pequeño que el anterior, debido a que $KG \in [0, 1]$. Cuanto mayor sea KG, más rápido encontrará el valor verdadero.

En cada iteración, algunos valores son obtenidos del proceso iterativo, los cuales serán usados para obtener un mejor resultado de aquello que se esté calculando. En el caso bajo estudio será la posición y la velocidad de un sistema UAV mediante simulación de la toma de datos de un GPS y una IMU. Cuantas más iteraciones se realicen, más precisos serán estos valores obtenidos.

4.3. El modelo multidimensional

En la sección anterior se introducen unas breves pinceladas de sobre qué trata el filtro de Kalman, así como las ecuaciones y cálculos básicos con el fin de entender el proceso. Pero el problema bajo estudio es más complicado que eso, puesto que es necesario tener en cuenta no sólo una medida en cada instante de tiempo, sino que se obtienen varios datos de entrada (tres medidas para las tres direcciones de la posición, otras tres para las de la velocidad, etc.). Por ello, es necesario explicar el modelo multidimensional de este tipo de filtros.

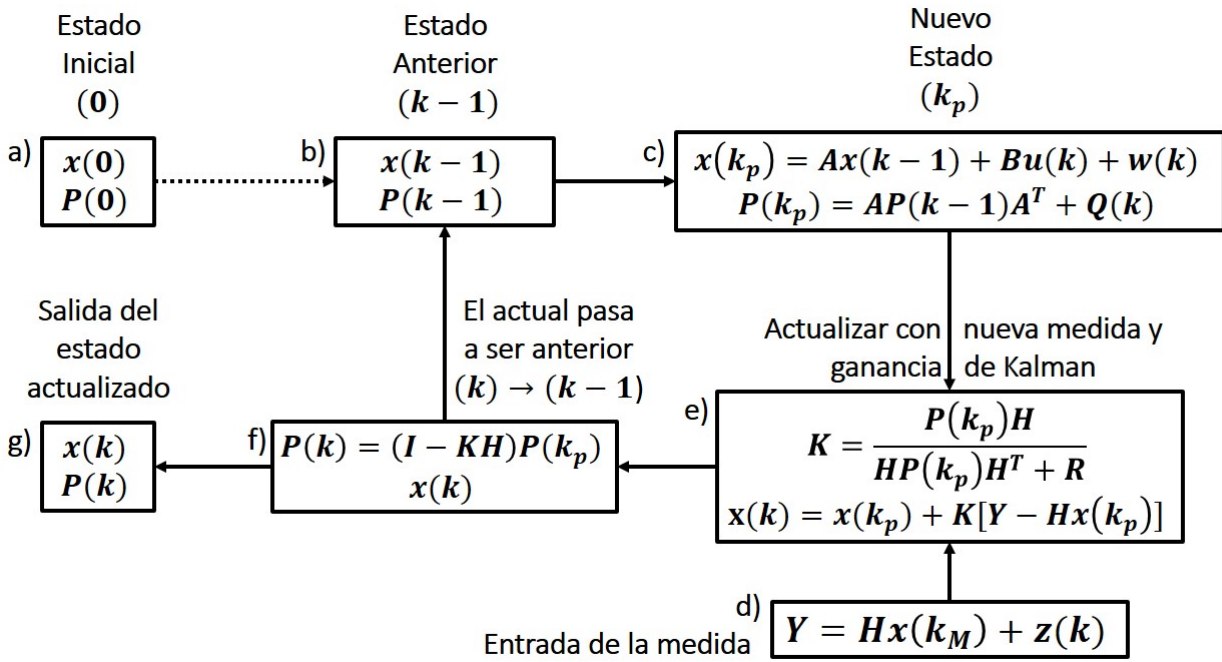


Figura 4.3: Proceso Iterativo que explica el modelo multidimensional mediante las ecuaciones del algoritmo

En la Figura 4.3 se intenta representar los diferentes pasos a seguir a la hora de implementar el filtro de Kalman mediante las ecuaciones que componen el algoritmo. Se intentará desarrollar cada uno de los recuadros señalados indicando qué significa cada variable y para posteriormente aclarar cómo se calcula cada una de ellas.

a) Estado Inicial. Estado desde el cual parte la simulación o el modelado del sistema. Consta de

- $\mathbf{x}(0)$ = Matriz de estados iniciales. En dicha matriz se incluirán los valores iniciales de las variables que se quieren filtrar. En el caso del movimiento de un vehículo, normalmente se estudiarán la posición y la velocidad, y por lo tanto, esta matriz constará de 6 elementos: las 3 coordenadas en posición y las 3 coordenadas en velocidad.
- $\mathbf{P}(0)$ = Matriz de la covarianza inicial del proceso. Hace referencia a los **errores en las estimaciones**, de los cuales se habló anteriormente. Este es el término que corresponde con el numerador de la ecuación 4.1. Posteriormente se explicará cómo se construye esta matriz.

b) Estado Anterior. En la primera iteración el estado inicial se convierte en el estado anterior. Una vez que se comience a realizar el proceso completo, este estado anterior vendrá definido por los cálculos propios del estado en un instante (actual) y no del inicial. En este punto las matrices $\mathbf{x}(k-1)$ y $\mathbf{P}(k-1)$ significan lo mismo que en el paso anterior, matriz de estados y matriz de covarianza del proceso (error en la estimación).

c) Nuevo Estado. A estas alturas ya se puede *predecir* el estado actual k_p en el que se encuentra el sistema. Las ecuaciones correspondientes a este punto son las siguientes

$$\begin{aligned} x(0) &\implies x(k-1) \implies x(k_p) = Ax(k-1) + Bu(k) + w(k) \\ P(0) &\implies P(k-1) \implies P(k_p) = AP(k-1)A^T + Q(k) \end{aligned} \quad (4.4)$$

donde

- $\mathbf{x}(k)$ = Matriz de estados del sistema en el instante actual k (posición actual, velocidad actual, o cualquier otra variable que se esté considerando). Esta primera ecuación de la expresión 4.4 se denomina “*Ecuación de estado*”.
- $\mathbf{u}(k)$ = Matriz de variables de control en el instante actual k . En el caso bajo estudio, donde se pretende controlar un vehículo que vuela al aire libre, una de esas variables de control podría ser por ejemplo la gravedad. Esta aceleración de la gravedad es constante y afecta directamente a la posición del vehículo en cada instante, por lo que sería necesario modelarla como una variable de control.
- $\mathbf{w}(k)$ = Matriz del ruido en el estado calculado. Como se ha hablado anteriormente, pasar de un modelo real a un modelo matemático implica errores de aproximaciones, redondeos o simplificaciones que deben ser tenidos en cuenta en algún punto. De ello se encarga esta variable $w(k)$. Cuanto más próximos estén el modelo simulado y el modelo real, menor será esta variable.
- $\mathbf{P}(k)$ = Matriz de la covarianza del proceso. La función es la misma que la explicada en el estado inicial: el error en la estimación calculada. Se verá cómo es necesario conocer la matriz A y su traspuesta para actualizarla en cada iteración. Esta depende de la covarianza calculada anteriormente y de la covarianza del ruido del proceso Q
- \mathbf{A}, \mathbf{B} = Matrices utilizadas para convertir el estado de entrada en el estado nuevo (A), y las variables de entrada en estado nuevo (B). Dependiendo del modelo que se realice del sistema será necesario utilizar unas matrices u otras. Más adelante se explicará también cómo se forman estas matrices dependiendo de las variables que se toman como estados y del modelo utilizado.
- \mathbf{Q} = Matriz de la covarianza del ruido del proceso. La función principal de Q es prevenir que la matriz P de covarianza del estado se convierta en un valor muy pequeño o eventualmente que llegue a cero. Si P llegase a valer cero, significaría que las medidas tomadas por los sensores que se están utilizando serían ignoradas por completo, aunque tuvieran una información de gran valor. [24]

En la primera ecuación de 4.4, se puede observar, por tanto, cómo para calcular el nuevo estado (por ejemplo, la nueva posición del objeto), es necesario conocer el estado anterior (la posición anterior), la aceleración que sufre el vehículo (o cualquier otra variable de control que afecte a su trayectoria), y el error producido por modelar un sistema real (si se utilizan las leyes de la Cinemática se desprecia el rozamiento, por ejemplo).

d) Entrada de la medida. Aquí entra en juego el papel de los **sensores**. Estos son los que toman las medidas y en este paso es en el que esas medidas se procesan para incluirlas en el proceso de filtrado de la señal. Para procesar esa medida es necesario tener en cuenta la siguiente expresión

$$y(k) = Hx(k_m) + z(k) \tag{4.5}$$

que es la denominada “*Ecuación de salida*”, y donde

- $\mathbf{y}(k)$ = Es el valor que se introduce en el proceso de filtrado, el cual corresponde a la medida tomada por el sensor $x(k_M)$, pero a la cual se le añade el correspondiente valor del error que comete el sensor $z(k)$. El nombre puede dar lugar a confusión, puesto que comúnmente se denomina ecuación de salida (salida del sensor, puesto que saca datos), pero que aquí se le ha nombrado como entrada de la medida (puesto que es el dato que entra en el filtro)
- $\mathbf{x}(k_M)$ = Medida (de ahí el subíndice k_M) tomada directamente por el sensor. Esta variable típicamente es un vector de datos, como por ejemplo, las 3 datos para la posición y 3 datos para la velocidad.

- \mathbf{H} = Es una matriz cuya función es la correspondiente con las matrices A y B de la ecuación 4.4. En estas ecuaciones a menudo es necesario coordinar que los estados (medidos por los sensores y estimados) se encuentran en la misma posición dentro de los vectores. Se verá más adelante qué forma debe tener esta matriz H para adaptar este vector de medidas a una salida con un formato determinado. Es también conocida como matriz \mathbf{C} .
- $\mathbf{z}(\mathbf{k})$ = Error en la medida, también denominado incertidumbre. Este es el error producido en el sensor al tomar los datos, del que se ha hablado en profundidad en el tema del modelado de sensores. La teoría del filtro de Kalman asume que el ruido del proceso \mathbf{w} y el ruido en la medida \mathbf{z} son independientes entre sí.

Nota: Una de las primeras medidas necesarias para eliminar el ruido de una señal, es que el proceso que se está midiendo debe poderse describir como un **sistema lineal**, de los cuales ya se ha hablado anteriormente. Muchos de los procesos físicos que ocurren a nuestro alrededor así como un vehículo circulando por la carretera, un satélite orbitando la Tierra, el eje de un motor movido por corrientes de aire, o una señal sinusoidal de radiofrecuencia pueden ser aproximados por sistemas lineales. ¿Qué implica ser un sistema lineal? En términos matemáticos, que el proceso pueda ser descrito por las ecuaciones 4.4 (Ecuación de estado) y 4.5 (Ecuación de salida) [20]. Estas ecuaciones son las mismas que las que se introdujeron al principio del tema del Modelado en Espacio Estados, las ecuaciones de 2.3, pero ya adaptadas al problema bajo estudio.

e) Actualizar con nueva medida y ganancia de Kalman. Una vez que se ha calculado la **estimación** teórica en **c)**, es el momento de incluirle un valor actual de las **medidas**, calculado en **d)** para actualizar los valores. Es decir, para simplificar: se han calculado ya de dos formas las variables (por ejemplo, posición). Se tiene la posición gracias al modelo matemático, y se tiene la posición gracias al dato del GPS/IMU que he recibido. ¿Cuál de los dos es más acertado? ¿Cuál tiene menos error? El que tiene la respuesta a esta pregunta, como se explicó en el apartado anterior, es el parámetro llamado ganancia de Kalman. Se calcula y utiliza como sigue

$$K = \frac{P(k_p)H}{HP(k_p)H^T + R} \quad (4.6)$$

$$x(k) = x(k_p) + K[y(k) - Hx(k_p)] = x(k_p) + K[Hx(k_m) - Hx(k_p) + z(k)]$$

donde

- \mathbf{K} = Ganancia de Kalman. Su función se ha explicado anteriormente, y de nuevo en la ecuación se puede observar que se trata de una comparativa entre un factor relacionado con el error de la estimación y otro factor relacionado con el error en la medida. En la segunda ecuación se observa cómo la ganancia de Kalman pondera, según los errores, entre la medida y la estimación.
- \mathbf{P} = Matriz de la covarianza del proceso. Explicada anteriormente.
- \mathbf{H} = Matriz de conversión igual a la explicada anteriormente. Se utiliza principalmente para conseguir las dimensiones adecuadas para poder realizar los cálculos correctamente.
- \mathbf{R} = Matriz de covarianza del ruido de los sensores. Esta matriz, como se verá posteriormente, deriva del error $z(k)$, el cual aparecía en la ecuación de salida y representaba el error del sensor.
- $\mathbf{x}(\mathbf{k})$ = Actualización de la matriz de estados. Esta actualización es la que tiene en cuenta la estimación y la medida tomada por los sensores. Analizando un poco más la expresión, se observa que, como se explicó también en el apartado anterior:
 - si la ganancia de Kalman es **alta** ($\mathbf{K}=\mathbf{1}$, error en estimación alto) \implies el término de la estimación $x(k_p) - KHx(k_p)$ se anula, y se tiene en cuenta el término de la medida $KHx(k_M)$.

- si la ganancia de Kalman es **baja** ($\mathbf{K}=\mathbf{0}$, error en medida alto) \implies el término de la medida $KHx(k_M)$ se anula, y se tiene en cuenta el término de la estimación $x(k_p) - KHx(k_p)$.

f) El actual pasa a ser anterior. En este punto, la estimación actual se convierte en la anterior, como es normal en cualquier proceso iterativo.. Se actualiza aquí también el valor del error de la estimación $P(k)$ para seguidamente procesar ese valor en la siguiente iteración, según la expresión

$$P(k) = (I - KH)P(k_p) \tag{4.7}$$

donde

- \mathbf{I} = Matriz identidad.
- \mathbf{P} , \mathbf{K} , \mathbf{H} = Las mismas que las descritas anteriormente.

En el recuadro del diagrama 4.2 se observa cómo de este punto se extraen tanto los datos para el siguiente apartado (salida del estado actualizado) como los datos para utilizarlos en los cálculos de la siguiente iteración (estado anterior).

g) Salida del estado actualizado. Estos datos de la matriz de estado y de la matriz de la covarianza del error. Gracias a estos datos ya se puede conocer la mejor aproximación de dónde se encuentra el vehículo en ese instante, su velocidad o cualquier otro dato que se haya considerado necesario para definir el estado.

Es el paso final del filtro de Kalman. Aquí se ha descrito un proceso el cual se ha considerado el mejor puesto que, como se dijo anteriormente, minimiza la varianza del error de la estimación. La aplicación inmediata que se le intentará dar en este trabajo será la de filtrar las señales procedentes de GPS y/o IMU, para obtener una mejor calidad de la información recibida.

Las principales ideas aquí descritas son realmente teóricas, será necesario aplicar dicho procedimiento al caso real bajo estudio para poder revelar la verdadera utilidad del Filtro de Kalman. A continuación se intentará desarrollar los pasos necesarios para construir cada variable que se ha descrito en este apartado, para ver cómo son en formato matricial y por qué, puesto que se considera que no hay mejor forma de explicar un concepto que ejemplificándolo.

5 | Caso práctico - Simulación en Matlab del Filtro de Kalman

Se intenta en este apartado esclarecer los conceptos tan teóricos que han sido descrito en el apartado anterior. Para ello, se ha realizado una simulación en Matlab para implementar un filtro de Kalman de manera práctica. Con ello, se intenta obtener un mejor resultado de una señal recibida mediante **GPS**, para el cálculo de una trayectoria rectilínea y uniforme. Puesto que el filtro de Kalman se usa para sistemas donde se reciben más de una señal aparte del GPS, se utilizará la herramienta **Google Maps** para modelar la trayectoria seguida por el vehículo, de forma que se tomará dicha trayectoria como si fuera otra señal (con todas sus consecuencias en cuanto a errores se refiere), para poder aplicar el filtro. A partir de los datos de ambas señales, GPS y Google Maps, se aplicará el algoritmo del filtro de Kalman para obtener una trayectoria estimada la cual tenga en cuenta los errores de ambas señales y así minimizarlo.

5.1. Procesamiento de datos

El programa utilizado para procesar los datos será Matlab. Se irán implementando las matrices, los vectores y las ecuaciones necesarias para ir tratando la información de manera ordenada, y se irán presentando los resultados que se van obteniendo. Se pretende en este apartado filtrar una señal recibida por GPS, por lo que se irá describiendo qué acciones se realizan para tal fin. En este caso, se quiere implementar un filtro de Kalman para una señal recibida por GPS dada una trayectoria rectilínea y a velocidad constante.

5.1.1. Medida de la trayectoria - GPS

La adquisición de datos para este problema se efectuará a través del receptor GPS incorporado en un *smartphone*. Para ello, se utilizará una aplicación para el seguimiento de la posición llamada “**NMEAGps**”. De ella, se obtendrá el archivo *nmea.log*, en el cual está contenida toda la información que el GPS manda, y que será procesada por Matlab.

Este archivo *nmea.log* contiene la información relevante sobre posición, velocidad, rumbo, tiempo, fecha, etc., en el formato ASCII. El protocolo utilizado es NMEA 0183, es un estándar que simplemente proporciona un formato determinado para la presentación de los datos. Para dar una idea más visual, un fragmento de este archivo se puede observar en la siguiente Figura 5.1.

```

$GPGSV,3,1,12,01,45,045,50,02,45,045,50,03,45,045,50,04,45,045,50*7C
$GPGSV,3,2,12,05,45,045,50,06,45,045,50,07,45,045,50,08,45,045,50*77
$GPGSV,3,3,12,09,45,045,50,10,45,045,50,11,45,045,50,12,45,045,50*71
$GPRMC,101857.00,A,3852.7021,N,658.8373,W,22.7,336.1,190816,0.0,E,A*16
$GPVTG,336.1,T,0.0,M,22.7,N,42.1,K*49
$GPGGA,101857.00,3852.7021,N,658.8373,W,12,12,1.0,173.5,M,0.0,M,0.0,0000*56
$GPGSA,A,3,01,02,03,04,05,06,07,08,09,10,11,12,1.2,1.0,0.6*35
$GPGSV,3,1,12,01,45,045,50,02,45,045,50,03,45,045,50,04,45,045,50*7C
$GPGSV,3,2,12,05,45,045,50,06,45,045,50,07,45,045,50,08,45,045,50*77
$GPGSV,3,3,12,09,45,045,50,10,45,045,50,11,45,045,50,12,45,045,50*71
$GPRMC,101858.00,A,3852.7079,N,658.8420,W,21.4,328.0,190816,0.0,E,A*1B
$GPVTG,328.0,T,0.0,M,21.4,N,39.7,K*4D
$GPGGA,101858.00,3852.7079,N,658.8420,W,12,12,1.0,173.5,M,0.0,M,0.0,0000*55
$GPGSA,A,3,01,02,03,04,05,06,07,08,09,10,11,12,1.2,1.0,0.6*35
$GPGSV,3,1,12,01,45,045,50,02,45,045,50,03,45,045,50,04,45,045,50*7C
$GPGSV,3,2,12,05,45,045,50,06,45,045,50,07,45,045,50,08,45,045,50*77
$GPGSV,3,3,12,09,45,045,50,10,45,045,50,11,45,045,50,12,45,045,50*71
$GPRMC,101859.00,A,3852.7139,N,658.8459,W,21.1,327.0,190816,0.0,E,A*1B
$GPVTG,327.0,T,0.0,M,21.1,N,39.1,K*41

```

Figura 5.1: Ejemplo del formato del archivo procesado NMEA

En la Figura de arriba solamente se han mostrado 19 líneas de código de un total de 4885. Puesto que este formato de representar una trayectoria de un vehículo no es la más figurativa, se ha decidido utilizar el programa Google Earth para que trace la línea seguida por el sistema, la cual se puede observar en la imagen de la Figura 5.2. Se ha utilizado ese programa puesto que es capaz de trabajar con archivos NMEA.log.



Figura 5.2: Seguimiento de la trayectoria gracias al programa Google Earth. *Recorrido por la ciudad de Badajoz*

Se decidió estudiar una trayectoria rectilínea y a velocidad constante, por lo que se utilizó la velocidad de cruce (velocidad constante) que proporcionan algunos automóviles, y a través de un puente (movimiento rectilíneo), en este caso el “*Puente de la Universidad*” de Badajoz. El archivo NMEA ahora utilizado ha pasado a tener 462 líneas de código, frente a las 4885 anteriores.

Dicho archivo tiene una serie de líneas de código diferentes como se pudo observar en la Figura 5.1. Para obtener los datos que son interesantes para el caso bajo estudio, es necesario saber qué información proporciona cada una de esas líneas (\$GPGSV, \$GPRMC, \$GPVTG, \$GPGGA...). Para conocer la

posición del vehículo es necesario conocer simplemente la estructura del código \$GPGGA, puesto que en ella aparecen las coordenadas Norte/Sur, Este/Oeste y altitud. Aquí se muestra un ejemplo de la estructura del código \$GPGGA, y en qué formato se encuentran los datos:

```
$GPGGA,101857.00,3852.7021,N,658.8373,W,12,12,1.0,173.5,M,0.0,M,0.0,0000*56
Tiempo  $\Rightarrow$  101857,00 = 10h18'57,00''
Latitud  $\Rightarrow$  3852,7021, N = 38°52,7021', Norte dotted
Longitud  $\Rightarrow$  658,8373, W = 6°58,8373', Oeste dotted
Altitud  $\Rightarrow$  173,5, M = 173,5 metros
```

Una vez que se ha obtenido el código correspondiente a la trayectoria rectilínea (Figura 5.3), se procesa el archivo con Matlab. Se puede observar en el apéndice B el código de Matlab utilizado para obtener dichos datos. En el archivo NMEA, la posición viene dada por (*latitud*, *longitud*, *altitud*). Se utiliza el código correspondiente a “**Cambio de (latitud, longitud, altitud) a (x, y z)**”, también adjuntado en dicho apéndice.



Figura 5.3: Trayectoria en línea recta y a velocidad constante. *Puente de la Universidad, Badajoz*

5.1.2. Medida de la trayectoria - Google Maps

Para obtener los datos de la trayectoria mediante Google Maps, se han tenido en cuenta los puntos iniciales y finales del puente, tomando su posición en (*latitud*, *longitud*, *altitud*), al igual que en el caso del GPS. Mediante la toma de esos puntos, se ha calculado el vector que contiene los datos de la trayectoria, usando el código de Matlab adjunto, en el **apéndice B**.

Es importante destacar que en un futuro, a la hora de representar los datos (tanto de Google Maps como de GPS), se utilizarán las **coordenadas (x, y, altitud)**, puesto que se ha considerado que sus gráficas son las más interesantes visualmente hablando.

Para explicar un poco el código, lo primero que se hace es cambiar de coordenadas (*latitud*, *longitud*, *altitud*) a (*x*, *y*, *z*). Se calcula la velocidad siguiendo las siguientes expresiones

$$\begin{aligned}
 x_{final} &= x_{inicial} + v_x t \\
 y_{final} &= y_{inicial} + v_y t \\
 altitud_{final} &= altitud_{inicial} + v_{altitud} t
 \end{aligned}
 \tag{5.1}$$

Puesto que la trayectoria bajo estudio es un puente, todos los puntos se encuentran a la misma altura con respecto al nivel del mar, y por ello, la altitud no varía (su velocidad es nula, $v_{altitud} = 0$).



Figura 5.4: Coordenadas de los puntos inicial y final del puente según Google Maps

5.2. Parámetros del filtro de Kalman

Puesto que ya se ha explicado en apartados anteriores lo que significa cada parámetro, aquí se irá mostrando cómo se forma cada uno de ellos. En el caso del estado del sistema, puesto que se quiere filtrar la posición, el estado del sistema constará de los valores $(x, y, altitud)$, tal y como se dijo anteriormente.

Para el cálculo de los ruidos del sistema, junto con las matrices de covarianza relacionadas con dichos ruidos, se han utilizado las siguientes expresiones, obtenidas de [11], y las cuales corresponden con las expresiones del apéndice A “*Varianza y Covarianza*”, pero ya aplicadas a este caso concreto.

- \mathbf{Q} = Matriz de covarianza del ruido del proceso. Como se puede observar en la ecuación 5.2, la matriz de covarianza del ruido depende de la diferencia entre el valor del error en el instante i y la media de dicho error, los cuales a su vez dependen de los valores de las variables de estado

x . Este hecho será importante para explicar el resultado obtenido al finalizar la simulación del filtro.

- \mathbf{w} = Error en el proceso.

$$Q = \frac{\sum_{i=0}^{n-1} (w_i - \bar{w})(w_i - \bar{w})^T}{n-1} \implies \text{donde}$$

$$\bar{w} = \frac{\sum_{i=0}^{n-1} (x_{t-1} - x_{t-i-1})}{n} \quad (5.2)$$

$$w_i = (x_{t-1} - x_{t-i-1})$$

- \mathbf{R} = Matriz de covarianza del ruido de los sensores.
- \mathbf{z} = Error en la medida.

$$R = \frac{\sum_{i=0}^{n-1} (z_i - \bar{z})(z_i - \bar{z})^T}{n-1} \implies \text{donde}$$

$$\bar{z} = \frac{\sum_{i=0}^{n-1} (y_{t-i} - H_{t-i}x_{t-i})}{n} \quad (5.3)$$

$$z_i = (y_{t-i} - H_{t-i}x_{t-i})$$

Los cálculos realizados en Matlab se pueden encontrar en el apéndice B. Según [11] y [24], a la matriz de covarianza del proceso \mathbf{P} se le puede asignar un valor cualquiera puesto que su valor se va actualizando en cada iteración del filtro y tiene la peculiar característica de que converge rápido a su valor real. También en esta literatura se indica que se utiliza una matriz diagonal, de forma que se imponga la no existencia de correlación entre las variables de estado. En el caso bajo estudio, esto significa que el desplazamiento producido en una variable de estado x no afecta al desplazamiento en las otras dos variables de estado y y *altitud*, y con estas dos el mismo caso. Con esto, se ha decidido asignar a \mathbf{P} el valor siguiente

$$P = \begin{bmatrix} 7 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 7 \end{bmatrix} \quad (5.4)$$

5.3. Resultados obtenidos de la simulación

Una vez que se han definido todos los parámetros para implementar el filtro de Kalman, se comienza con el proceso iterativo, tal y como se mostraba en el diagrama de la Figura 4.3. De esta forma, tal y como se puede vislumbrar en el código ejecutado en el apéndice B, se pueden obtener unos datos sobre una trayectoria estimada, basada en los datos obtenidos por el **GPS** y por **Google Maps**. El filtro de Kalman asegura obtener un resultado en el cual la varianza del error es la mínima, es decir, la **trayectoria estimada** una vez llevado a cabo el algoritmo del filtro será la más parecida a la **trayectoria real** (la que cuenta con menos error).

La Figura 5.5 se ha girado y colocado de forma que se vea lo más claro posible el efecto del filtro sobre las señales.

La gráfica en color verde discontinua se corresponde con los datos obtenidos del **GPS**. Como se puede observar, la altitud proporcionada por el GPS no es muy fiable, puesto que se conoce que la trayectoria calculada fue tomada en un puente, el cual no varía de altitud del orden de los 8 metros (de 166m a 174m) tal y como se ve en la figura.

La gráfica en color azul discontinua corresponde con los datos calculados a partir de **Google Maps**. Se observa claramente que se trata de un movimiento rectilíneo, puesto que la altura es completamente constante e igual a 167m. Ésta difiere un par de metros de los 169m que se obtienen de media si se toman los datos del GPS. Esta trayectoria se puede decir que es aquella que se intentaba obtener cuando se tomaron las muestras. Se pretendía circular a velocidad constante en un tramo donde la altura con respecto al nivel del mar fuese lo más constante posible, por lo que surgió la idea de tomar muestras sobre un puente.

¿Cuál es el problema? Que existen múltiples factores que introducen error en estos datos. Para empezar, es muy difícil controlar que la velocidad de un vehículo sea completamente constante. Por ello se decidió utilizar un coche que tuviese la opción de conducir con velocidad de cruce, pero aun así no se asegura que dicha velocidad no varíe en ningún punto. A la hora de controlar el coche, también una fuente de error reside en los pequeños desplazamientos que se producen por movimientos en el volante. También es posible que se desvíe de la trayectoria ideal debido a fuerzas externas, como pueden ser la acción del viento o irregularidades en el terreno. Y por supuesto, los valores han sido tomados por una herramienta como Google Maps, la cual no se libra de errores a la hora de proporcionar la posición del punto que se señale en el mapa. Con esto se pretende decir que esta “señal ficticia” dista también, aunque en menor medida, de la **trayectoria real** seguida por el vehículo.

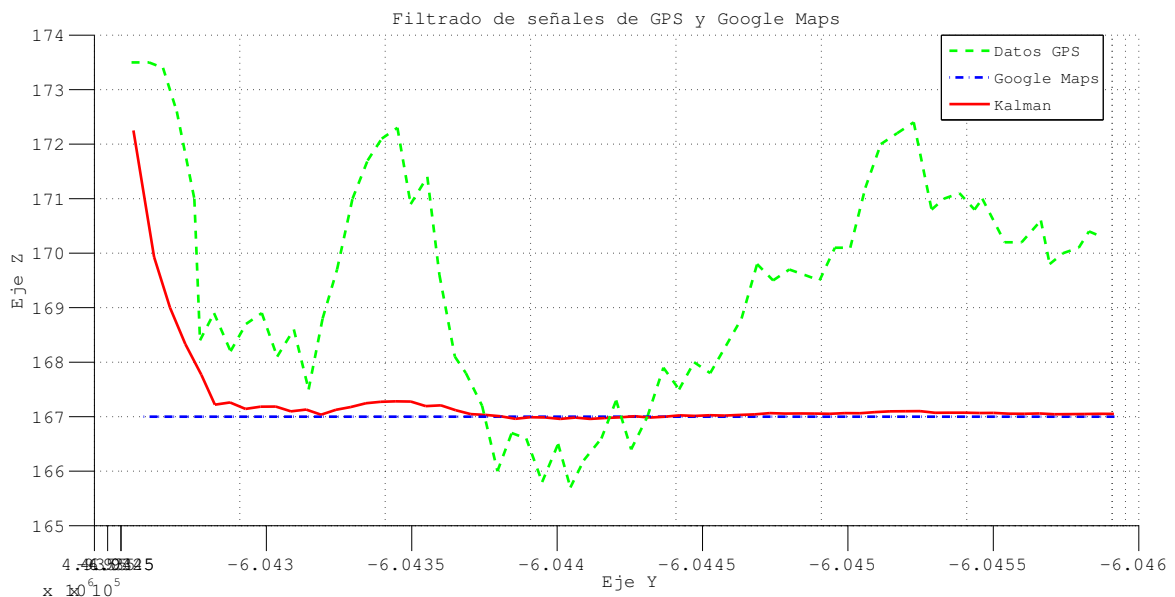


Figura 5.5: Comparación de las señales GPS, Google Maps y filtro de Kalman en altitud

Obtener una trayectoria que diste lo menos posible de la **trayectoria real** es el objetivo por el que se ha realizado esta simulación. Intuitivamente, la trayectoria real que se ha seguido con el vehículo será bastante parecida a la trayectoria calculada por Google Maps, en el sentido de que sería muy similar a una línea recta, puesto que las desviaciones producidas por los factores de error explicados en el apartado anterior son pequeños en este caso. En el caso de estudiar un UAV de verdad, dichos factores cobrarían más importancia debido a que son más sensibles a perturbaciones externas, debido a su menor peso, tamaño y estabilidad. Es decir, es más fácil que se desvíe un UAV 3 metros debido a ráfagas de aire que a un coche. Dicha trayectoria casi rectilínea sería interesante dibujarla también en la Figura 5.5, para de verdad comprobar lo que se pretende en esta simulación: que la trayectoria de Kalman es la más acertada a la hora de estimar una trayectoria real.

Centrados de nuevo en la Figura 5.5, se observa cómo la **trayectoria estimada** por el filtro de Kalman (en color rojo, continua) se encuentra en todo momento en un punto intermedio entre la señal de **GPS** y la señal de **Google Maps**. Se recuerda que esto es debido a que la ganancia de Kalman era un factor que ponderaba la importancia relativa que se le debe dar a una señal con respecto a la otra, basándose en el error en cada momento. Es decir, en cada iteración cuando se obtienen nuevos

datos, la ganancia de Kalman hará que el nuevo dato de la estimación conste de un porcentaje de una señal (por ejemplo, ganancia de Kalman=0.6 implica que tomará un 60% de la señal GPS) y de otro tanto por ciento de la otra (en este caso tendría un 40% de la señal de Google Maps).

Se observa cómo al principio de la gráfica la trayectoria de Kalman se encuentra en un valor más cercano al GPS y luego se aproxima cada vez más a la señal de Google Maps. El inicio de la trayectoria de Kalman es tan inexacto debido a que se definió en primer lugar una **P aleatoria y de valor alto** para, de hecho, comprobar que dicha matriz converge y se va acercando cada vez más al valor que posee menor error, en este caso la trayectoria de Google Maps. Para visualizar mejor este hecho, la Figura 5.6 se corresponde con una matriz P que en lugar de ser una diagonal de valor 7, es una diagonal de valor 1. Se observa cómo la gráfica de la trayectoria de Kalman comienza más próxima al valor más acertado, en este caso la señal de Google Maps, la cual posee menos errores que la de GPS (en altitud).

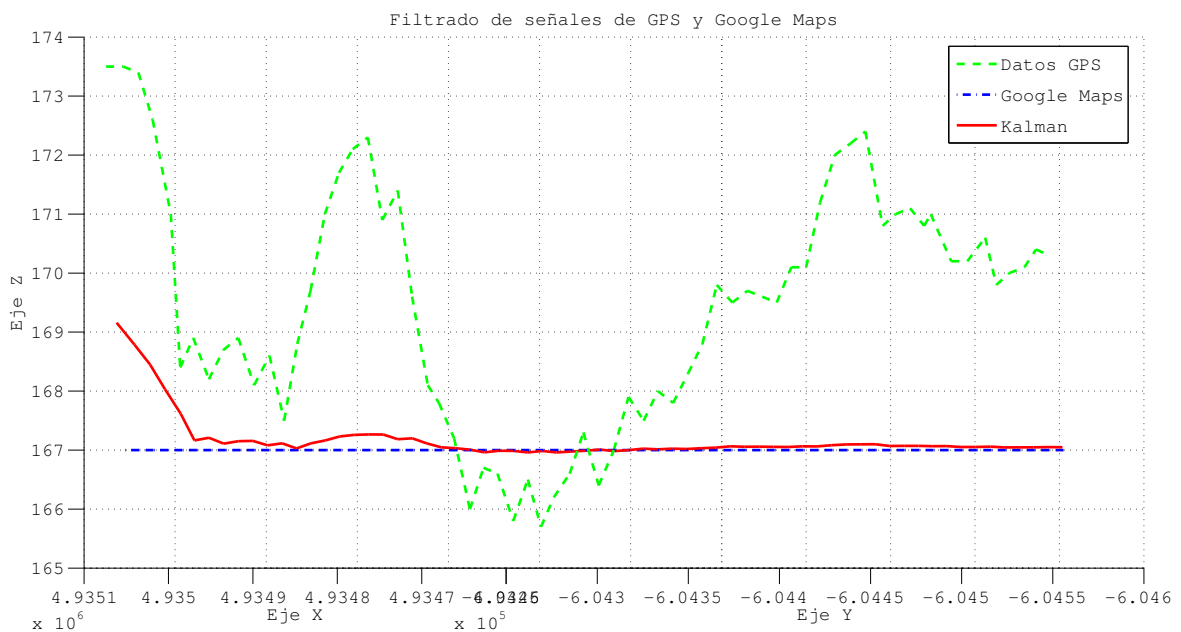


Figura 5.6: Comparación de las señales GPS, Google Maps y filtro de Kalman en altitud, **P más baja**

Anteriormente se dijo que sería interesante el valor de la matriz Q para interpretar el resultado, y aquí se ve claramente por qué: se observa cómo rápidamente el filtro decide que debe acercarse más a la trayectoria de Google Maps, por lo tanto, considera que dicha medida posee menor error que la medida del GPS. Puesto que los errores residen en las matrices Q y R, habrá que analizarlas para encontrar el por qué de este fenómeno. Pues bien, comparando ambas, se observa que la matriz Q tiene la siguiente forma

$$Q = \begin{bmatrix} * & * & 0 \\ * & * & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.5)$$

Este fenómeno se debe a que los errores dependen de cuánto varían las medidas con respecto a su valor medio (así se calcula la covarianza, ecuación 5.2), y se sabe que la altitud calculada según Google Maps es constante, por lo que dicha diferencia será nula. Este hecho de tener un valor nulo hace que la ganancia de Kalman siempre se vaya a decantar un poco más por este valor que por otro que sí tenga dicho error, por muy pequeño que sea será mayor que éste.

Parece que, puesto que el GPS proporciona medidas bastante erróneas de altitud, que no se debería confiar en el valor proporcionado por dicho sistema. Pero en la Figura 5.7, se ha intentado orientar la figura para que se vea en los ejes (x, y), tal y como se veía en la imagen de Google Earth 5.3.

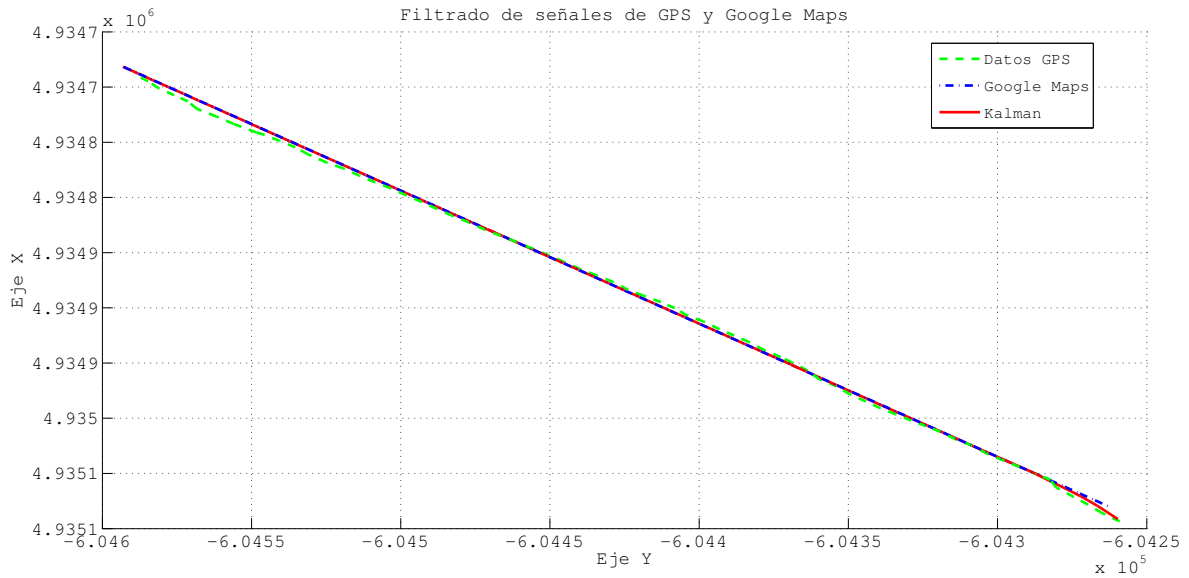


Figura 5.7: Comparación de las señales GPS, Google Maps y filtro de Kalman en ejes (x,y)

En esta figura ya es más difícil diferenciar qué señal se encuentra más próxima a la trayectoria estimada por el filtro de Kalman. También parece ser que desde esta perspectiva se aprecia que sigue mejor a la señal de Google Maps, pero en este caso no dista tanto de la señal de GPS. En este caso sí que se podría afirmar que GPS proporciona datos bastante fiables, acertados con la posición “real”, en el caso de que la señal de Google Maps fuera completamente la real. En cualquier caso, si se quisiera conocer con **mayor exactitud** el camino real seguido por el vehículo, habría que mirar la gráfica de color rojo, la cual se corresponde con la **trayectoria estimada por el filtro de Kalman**, puesto que es la que menos errores posee.

6 | Conclusiones y Líneas Futuras

Con este estudio se ha intentado analizar el estado actual en el que se encuentran los sensores y los sistemas que integran un UAV. Para comenzar el proyecto, se examinó la herramienta de espacio estados, puesto que se considera el instrumento más útil a la hora de trabajar con sistemas multivariantes, como pueden ser los vehículos aéreos no tripulados.

Con el estudio sobre el espacio de estados, se obtiene una visión general de cómo plantear un problema real de seguimiento de trayectorias, es decir, qué variables se deben tomar como estados puesto que definen el sistema modelado. Para ello, fue necesario introducir un apartado sobre los diferentes tipos de sensores que se encuentran implementados en los drones. Se obtiene de este apartado que el desarrollo de esta tecnología tan puntera, no depende sólo de un ámbito de la ingeniería como puede ser la aeroespacial, sino que se trata de un entorno en el que se necesita trabajar con grupos de ingenieros de muchas especialidades, como pueden ser automáticos, electrónicos, de aviónica, mecánicos, de energía, etc.

Al estudiar dichos sistemas, se observó que este sector de los UAVs está bastante avanzado en el ámbito militar, como es lógico, puesto que allí se desarrollaron en primer lugar antes de salir al mercado civil. De hecho, muchas veces, a la hora de indagar más a fondo sobre el funcionamiento de algunos sensores aplicados en estos vehículos, no se disponía de la información suficiente para conocer en detalle las características del funcionamiento del sistema, debido a la confidencialidad tanto de empresas como de militares.

En este caso de los sensores, son fundamentales los sistemas de navegación como pueden ser la IMU y el GPS. Se puede decir que el resto de sensores son más bien opcionales, los cuales se usarán cuando se diseñe un dron específico para unos fines específicos. Pero los sistemas de navegación tanto inerciales como por satélites son, como ya se ha dicho anteriormente, un “*must*” dentro del mundo de los UAVs.

Es verdad que la tecnología avanza a pasos agigantados, pero en este estudio se ha comprobado que existen dispositivos muy utilizados a día de hoy con problemas de precisión. Por tanto, es necesario seguir trabajando para conseguir reducir aún más sus fuentes de error para obtener datos que sean lo más precisos y seguros posible. No solamente se necesita desarrollar en particular la estructura y/o diseño de un sistema, sino que se deben desarrollar métodos de integración de sistemas que sean más fiables, capaces de minimizar los errores y que se aproximen lo más posible a la realidad que se desea medir.

La actual implementación de nuevos sensores y sistemas en distintos dispositivos gracias a los avances tecnológicos hace necesario que el estudio y desarrollo de la integración de señales no se quede atrás. Todavía existen numerosos aspectos en los que el filtro de Kalman puede ser mejorado, es por ello que a día de hoy se siguen publicando estudios en los cuales se proponen nuevos métodos de implementación de dicho filtro. Es necesario conocer correctamente los principios en los que se fundamenta el algoritmo utilizado por este filtro, para así vislumbrar los posibles errores que se comete al realizar el proceso iterativo y así, poder mejorarlo desarrollando una técnica alternativa.

Se ha observado también la facilidad con la que se encuentra actualmente una amplia gama de componentes electrónicos, desde baratos que te ofrecen un servicio básico hasta los más caros que cuentan con unos rangos de errores bastante reducidos y abarcan un mayor espectro dentro de las actuaciones que puede llevar a cabo. De hecho, mismo en los *smartphones* que se utilizan cada día, se encuentran por ejemplo el GPS, acelerómetros, giróscopos y magnetómetros. Con esto se pretende señalar que los sistemas de navegación no son algo que se encuentre fuera de nuestro alcance, sino que

además estamos en contacto con ellos a diario.

En cuanto al caso práctico llevado a cabo en este estudio del Trabajo de Fin de Grado, se han encontrado dificultades al intentar materializar los conceptos teóricos. La literatura que se ha utilizado para entender los algoritmos en los que se basa el filtro de Kalman es bastante confusa. Ha sido todo un reto entender la información encontrada, descubriendo que para el desarrollo de problemas parecidos, existen diversas formas de interpretar el filtro de Kalman. Se recomienda que en estudios posteriores, se detalle en profundidad la funcionalidad de cada parámetro utilizado, y por qué adopta la forma que tiene.

También se ha observado claramente que el GPS no proporciona unos datos demasiado fiables en cuanto a la variable altitud, tal y como se puede apreciar en las gráficas representadas en el apartado del Caso Práctico. La dependencia que presenta este sistema de una organización estadounidense, anima al resto de la comunidad tecnológica a desarrollar sus propios métodos de navegación basados en GNSS, como son la europea con Galileo, la rusa con GLONASS, la china con BeiDou, etc. La integración de varios sistemas de este tipo aumentaría notablemente la calidad de la señal recibida, y por tanto, la fiabilidad de la información, puesto que a día de hoy se encuentra muy sujeta a las decisiones que tomen las autoridades estadounidenses sobre este ámbito.

Como líneas futuras para este trabajo, sería interesante implementar el filtro de Kalman para señales procedentes no solamente del GPS, sino también de una IMU. Como se dijo en el apartado del estudio sobre el GPS, es recomendable, cuando un vehículo se encuentra navegando, disponer de sistemas de medidas inerciales aparte de sistemas GNSS. De hecho, la IMU es el sistema inercial más extendido dentro del ámbito de los UAVs. Típicamente los drones que se encuentran en el mercado cuentan con **GPS+IMU**, por lo tanto, aplicar un filtro de Kalman a las señales recibidas por dichos sistemas sería un problema que despierta bastante interés por el extendido uso que se le podría dar.

De esta forma, sería posible estudiar el caso en el que ambas señales son obtenidas de sensores reales e integradas. Es decir, la señal con la que se ha trabajado en este estudio obtenida mediante Google Maps sería sustituida por una real. En ese caso, la simplificación de trayectoria rectilínea y uniforme llevada a cabo en este documento no sería necesaria.

En el presente trabajo, se ha simulado en Matlab un filtro de Kalman sencillo para esclarecer los conceptos que se utilizan en dicho algoritmo, pero existen otros tipos de filtros también utilizados en la literatura relacionada. Por ejemplo, [6] propone utilizar un método denominado “*Square Root Cubature Kalman Filter, CKF*” con el fin de reducir los problemas relacionados con el filtro de Kalman usual: cantidad de cálculos a realizar, complejidad de las fórmulas y las transformaciones, “baja” exactitud, poca convergencia o incluso divergencia. Mediante su estudio, se demuestra que utilizando este método se obtienen mayores estabilidad y precisión que utilizando el filtro de Kalman y otra modalidad llamada “*Unscented Kalman Filter, UKF*”. También según [6], este último UKF atrae la atención en este área puesto que obtiene una media y una covarianza usando un muestreo determinista, pero como inconveniente pero tarda más tiempo que por ejemplo el “*Extended Kalman Filter, EKF*” [23]. Además, cuando el número de estados es mayor de 3, el UKF no funciona correctamente, incluso puede llegar a detener su operación. Se propone estudiar alguna de estas variantes del filtro de Kalman para implementarla en un sistema que integre señales IMU y GPS.

Como se ha recalado varias veces durante este trabajo, la navegación de cualquier tipo de vehículo es normalmente un problema de estimación de estado altamente no lineal. El EKF ha jugado un papel importante en dicha estimación durante décadas [1]. Sería interesante intentar implementar estas versiones del filtro de Kalman y comparar los resultados con los obtenidos con otros métodos de filtrado.

Apéndice A

Varianza & Covarianza

- Medida individual: x_i
- Media de las medidas: \bar{x}
- Desviación de la media: $\bar{x} - x_i$
- Cuadrado de la desviación: $(\bar{x} - x_i)^2$
- Varianza: $\sigma_x^2 = \frac{\sum_{i=1}^N (\bar{x} - x_i)^2}{N}$
- Covarianza: $\sigma_x \sigma_y = \sum_{i=1}^N (\bar{x} - x_i)(\bar{y} - y_i)$
- Desviación Estándar: $\sigma_x = \sqrt{\sigma_x^2} = \sqrt{\frac{\sum_{i=1}^N (\bar{x} - x_i)^2}{N}}$
- $2D = \begin{bmatrix} \sigma_x^2 & \sigma_x \sigma_y \\ \sigma_y \sigma_x & \sigma_y^2 \end{bmatrix}$
- $3D = \begin{bmatrix} \sigma_x^2 & \sigma_x \sigma_y & \sigma_x \sigma_z \\ \sigma_y \sigma_x & \sigma_y^2 & \sigma_y \sigma_z \\ \sigma_z \sigma_x & \sigma_z \sigma_y & \sigma_z^2 \end{bmatrix}$

Apéndice B

Código Matlab de caso práctico

- Implementación del filtro de Kalman

```
% Jose Luis Carretero Rodriguez
% Trabajo Fin de Grado - version 3D
% Seal GPS/Google Maps con Filtro de Kalman
% Nomenclatura:
% LLH = Latitud, Longitud, Altitud
% [x;y;vect_alt] = datos GPS
% [x_est;y_est;altitud_est] = datos GoogleMaps
% [vect_x_kalman;vect_y_kalman;vect_alt_kalman] = datos filtro
clear; clc; format long

%% Iniciar variables
cont=0; any_south_lat=0; any_west_long=0;
vector_lat=[]; vector_long=[]; vect_alt=[];
x=[];y=[];z=[];
velocidad=[];
vector_kalman=[];

% Modelo de la Tierra WGS-84
a=6378137;
f=1/298.257223563;

%% Obtengo datos del fichero NMEA
TLINE = fopen('puente1.log');
tline = fgets(TLINE);
while ischar(tline);
    token=strtok(tline);
    % VELOCIDADES
    % Obtengo la velocidad a partir de la linea GPVTG
    if strcmp(tline(2:6), 'GPVTG')==1
        [~,~,~,~,~,~,speed]=NMEA_VTG(tline);
        % Compruebo que no hay errores en el mensaje
        flag=checksum(token);
        if flag==0
            disp('Error') % Si hay error, muestro por pantalla
        else
            % Convierto la cadena a nmero
            speed=str2num(speed);
            % Almaceno las velocidades
            velocidad=[velocidad; speed]; % Si no hay error, guardo
        end
        % LATITUD LONGITUD ALTITUD
        % Obtengo la posicion a partir de la linea GPGGA
    elseif strcmp(tline(2:6), 'GPGGA')==1
        flag=checksum(token);
        if flag==0
            disp('Error')
        end
        [utc,lat,NS,long,EW,~,~,~,alt,unit_alt]=NMEA_GGA(tline);
        alt;
        if strcmp(NS, 'S') % Si estoy en el Sur, mostrar
            %disp('Hay sur') % Nunca estoy en el Sur en este caso
            lat=num2str(-str2double(lat));
            any_south_lat=1;
        end
        if strcmp(EW, 'W') % Si estoy en el Oeste, mostrar
            %disp('Hay Oeste') % Siempre estoy en el Oeste en este caso
            long=num2str(-str2double(long)); % Cambiar signo
            any_west_long=1;
        end
    end
end
```

```
end

% Guardo todas las medidas de posicion. LLH
% Formato string
vector_lat=[vector_lat; lat];
if size(long)~=9
    long=strcat(long,'0'); % dimensiones correctas
end
vector_long=[vector_long; long];
% Formato numero
vect_alt=[vect_alt; str2double(alt)];
end
tline=fgets(TLINE); % Obtener nueva linea
end

% Convertir lat y long de string a numero
for i=1:length(vector_lat)
    vect_lat(i,1)=str2double(vector_lat(i,1:2))+...
        str2double(vector_lat(i,3:end))/60;
    vect_long(i,1)=str2double(vector_long(i,2))+...
        str2double(vector_long(i,3:end))/60;
end

% Convertir LLH->xyz
for i=1:length(vector_lat)
    xyz=llh2xyz([vect_lat(i) -vect_long(i) vect_alt(i)],a,f);
    % Datos de posicin del GPS (x,y,z)
    x=[x; xyz(1)];
    y=[y; xyz(2)];
    z=[z; xyz(3)];
end

% Calcular altitud media
alt_media=sum(vect_alt)/66;

% plot(x)
% plot(y)
% plot(vect_alt); hold on
% plot(alt_media*ones(66,1))

% plot3(x,y,z,'LineWidth',2),grid
% xlabel('Eje X')
% ylabel('Eje Y')
% zlabel('Eje Z')

% Matriz de covarianza R - error en GPS
% Primero calculo la media del error z
sumatorio_GPS=[0;0;0];
for i=1:65
    sumatorio_GPS=sumatorio_GPS+...
        [x(i+1)-x(i);...
        y(i+1)-y(i);...
        vect_alt(i+1)-vect_alt(i)];
end
media_z=sumatorio_GPS/66;
% Calculo R
R1=0*ones(3);
for i=1:65
    R1=R1+[x(i+1)-x(i)-media_z(1);...
        y(i+1)-y(i)-media_z(2);...
        vect_alt(i+1)-vect_alt(i)-media_z(3)];*...
        [x(i+1)-x(i)-media_z(1);...
        y(i+1)-y(i)-media_z(2);...
        vect_alt(i+1)-vect_alt(i)-media_z(3)];';
end
R=R1/65; % 65 segundos dura la trayectoria - 65 muestras.
```



```

%% Google Maps
% Puente segun Google Maps (lat,long,alt)
lat_ini_est=38.878408; long_ini_est=-6.980620;
lat_fin_est=38.883509; long_fin_est=-6.984960;
alt_est=167;

% Convierto a (x,y,z)
est_ini=llh2xyz([lat_ini_est long_ini_est alt_est],a,f);
x_ini_est=est_ini(1);
y_ini_est=est_ini(2);

est_fin=llh2xyz([lat_fin_est long_fin_est alt_est],a,f);
x_fin_est=est_fin(1);
y_fin_est=est_fin(2);

% Calculo velocidades - (final-inicial)/tiempo
velocidad_x=(x_fin_est-x_ini_est)/65;
velocidad_y=(y_fin_est-y_ini_est)/65;

% Creo vector de componentes de posicion
for t=1:66
    x_est(t)=x_ini_est+velocidad_x*t;
    y_est(t)=y_ini_est+velocidad_y*t;
    altitud_est(t)=alt_est+0*t;
end
x_est=x_est';
y_est=y_est';
altitud_est=altitud_est';

% Matriz de covarianza Q
% Primero el ruido medio de w
sumatorio_est=[0;0;0];
for i=1:65
    sumatorio_est=sumatorio_est+...
        [x_est(i+1)-x_est(i);...
        y_est(i+1)-y_est(i);...
        altitud_est(i+1)-altitud_est(i)];
end
media_w=sumatorio_est/66;
% Matriz Q
Q1=0*ones(3);
for i=1:65
    Q1=Q1+...
        [x_est(i+1)-x_est(i)-media_w(1);...
        y_est(i+1)-y_est(i)-media_w(2);...
        altitud_est(i+1)-altitud_est(i)-media_w(3)]*...
        [x_est(i+1)-x_est(i)-media_w(1);...
        y_est(i+1)-y_est(i)-media_w(2);...
        altitud_est(i+1)-altitud_est(i)-media_w(3)]';
end
Q=Q1/65;

% figure('Name','Posicin x')
% plot(x_est,'r'); hold on
% plot(x,'b')
%
% figure
% plot(y_est,'r'); hold on
% plot(y,'b')
%
% figure
% plot(x,y,'r'); hold on
% plot(x_est,y_est,'b')
% xlabel('Eje X')
% ylabel('Eje Y')

```

```

% figure
% plot3(x,y,vect_alt,'g','LineWidth',2); grid; hold on
% plot3(x_est,y_est,altitud_est,'b','LineWidth',2)
% xlabel('Eje X')
% ylabel('Eje Y')
% zlabel('Altitud')
% title('Comparacin GPS con Estimada')

%% Filtro de Kalman

A=eye(3);
H=eye(3);

P=diag([7, 7, 7]); % [Lynch] - Error converge, "no importa" inicio

for k=1:66 % es 65

    % Matriz de Covarianza. Error en Estimacion
    P=A*P*A'+Q;

    % Ganancia de Kalman
    K=(P*H)/(H*P*H'+R);

    % Nuevo dato del GPS
    GPS=[x(k); y(k); vect_alt(k)];

    % Nuevo dato de Google Maps
    EST=[x_est(k); y_est(k); altitud_est(k)];

    % Actualizar Estado CON KALMAN
    estados_k=EST+K*(GPS-H*EST);
    if k==1 % Primera iteracion - creo variable
        vect_x_kalman=estados_k(1);
        vect_y_kalman=estados_k(2);
        vect_alt_kalman=estados_k(3);
    elseif k~=1 % Siguietes - amplio vector
        vect_x_kalman=[vect_x_kalman; estados_k(1)];
        vect_y_kalman=[vect_y_kalman; estados_k(2)];
        vect_alt_kalman=[vect_alt_kalman; estados_k(3)];
    end

    % Actualizar Matriz P
    P=(eye(3)-K*H)*P;
end

%% Display

figure('Name','Comparacin seales')
plot3(x,y,vect_alt,'g--','LineWidth',2); grid; hold on
plot3(x_est,y_est,altitud_est,'b-.','LineWidth',2); hold on
plot3(vect_x_kalman,vect_y_kalman,vect_alt_kalman,'r','LineWidth',2); hold on

title('Filtrado de una seal de GPS')
xlabel('Eje X')
ylabel('Eje Y')
zlabel('Eje Z')
legend('Datos GPS','Google Maps','Kalman')

fclose('all');

```

- Cambio de (latitud, longitud, altitud) a (x, y, z)

```
%llh=[latitud,longitud,altitud]
```

```

%Cuidado con radianes
%Cuidado con OESTE Y SUR!!!!

function xyz=llh2xyz(llh,a,f)
coslat=cos(llh(:,1)*pi/180); %coseno de latitud
sinlat=sin(llh(:,1)*pi/180); %seno de latitud
RN=a./sqrt(1-f*(2-f)*sinlat.^2);

xyz(:,1)=(RN+llh(:,3))*coslat.*cos(llh(:,2)*pi/180);
xyz(:,2)=(RN+llh(:,3))*coslat.*sin(llh(:,2)*pi/180);
xyz(:,3)=(1-f)^2*RN+llh(:,3)).*sinlat;

return
end

```

- Comprobación de errores en la señal GPS. Método basado en la norma seguida por el formato NMEA.

```

function flag=checksum(message)
[str,remain]=strtok(message,'$*');
remain=remain(2:end);
str=uint8(str);
chk=0;
for i=1:size(str,2)
    chk=bitxor(chk,str(i));
end
chk=dec2hex(chk);
if strcmp(remain,num2str(chk))==1
    flag=1;
else
    flag=0;
end
end

```

- Obtención de los datos a partir de las líneas de código NMEA: líneas GPGGA y GPVTG

```

function [utc,lat,NS,long,EW,qual,nsat,...
    HDOP,alt,unit_alt,height,unit_height,age,ID]=NMEA_GGA(message)

if strcmp(message(2:6),'GPGGA')==1 %Si es el mensaje correcto
    remain=message(8:end);
    [utc,remain]=NMEA_get_field(remain,',');
    [lat,remain]=NMEA_get_field(remain,',');
    [NS,remain]=NMEA_get_field(remain,',');
    [long,remain]=NMEA_get_field(remain,',');
    [EW,remain]=NMEA_get_field(remain,',');
    [qual,remain]=NMEA_get_field(remain,',');
    [nsat,remain]=NMEA_get_field(remain,',');
    [HDOP,remain]=NMEA_get_field(remain,',');
    [alt,remain]=NMEA_get_field(remain,',');
    [unit_alt,remain]=NMEA_get_field(remain,',');
    [height,remain]=NMEA_get_field(remain,',');
    [unit_height,remain]=NMEA_get_field(remain,',');
    [age,remain]=NMEA_get_field(remain,',');
    [ID,remain]=NMEA_get_field(remain,'*');
end

```

GPVTG

```

function [TC,unit_TC,MC,unit_MC,...
    speed_kts,unit,speed_kmh,unit2,mode]=NMEA_VTG(message)

```

```
if strcmp(message(2:6), 'GPVTG')==1 %Si es el mensaje correcto
    remain=message(8:end);
    [TC, remain]=NMEA_get_field(remain, ',');
    [unit_TC, remain]=NMEA_get_field(remain, ',');
    [MC, remain]=NMEA_get_field(remain, ',');
    [unit_MC, remain]=NMEA_get_field(remain, ',');
    [speed_kts, remain]=NMEA_get_field(remain, ',');
    [unit, remain]=NMEA_get_field(remain, ',');
    [speed_kmh, remain]=NMEA_get_field(remain, ',');
    [unit2, remain]=NMEA_get_field(remain, '*');
    %[mode, remain]=NMEA_get_field(remain, '*');
end
```

Para obtener dato dentro de la línea

```
function [field, remain]=NMEA_get_field(str, delimiter)
field=[];
remain=str;
if isempty(str)
    error=1;
end
if strcmp(str(1:2), [' ', delimiter]);
    remain=str(2:end);
else [field, remain]=strtok(str, [' ', delimiter]);
end
return
```

Bibliografía

- [1] Nobe S A and WANG Fei-yue. An overview of recent developments in automated lateral and longitudinal vehicle controls. *IEEE International Conference on SMC. Tucson, AZ*, pages 3447–3452, 2001.
- [2] Remus C. Avram, Xiaodong Zhang, Jacob Campbell, and Jonathan Muse. IMU Sensor Fault Diagnosis and Estimation for Quadrotor UAVs. *Elsevier Ltd.*, 48(21):380–385, 2015.
- [3] Randal W. Beard and Timothy W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.
- [4] John Byron Buitrago. *Representación en el espacio de estado*. Departamento de Ingeniería Electrónica, Universidad de Antioquia, EAFIT, Medellín, Colombia, 2010-2012.
- [5] Miguel Martínez de Lahidalga Martínez. *PFC: Desarrollo de hardware embarcado en UAV*. Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla, 2015.
- [6] Jianmin Duan, Hui Shi, Dan Liu, and Hongxiao Yu. Square Root Cubature Kalman Filter-Kalman Filter Algorithm for Intelligent Vehicle Position Estimate. *Elsevier Ltd.*, 137:267–276, 2016.
- [7] Hosam K. Fathy, Zoran S. Filipi, Jonathan Hagen, and Jeffrey L. Stein. Review of hardware-in-the-loop simulation and its prospects in the automotive area. *Modeling and Simulation for Military Applications*, 6228, 2006.
- [8] Gene F. Franklin, J. David Powell, and Michael L. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley Publishing Company, segunda edición, 1990.
- [9] Gabriel Infante, Jeremías Báez Carballo, Federico Díaz Báez, and Cristian David Cavenio. *Adición y Calibración de Sensores para un Cuadrorotor*. Departamento de Ingeniería Electrónica, Universidad Tecnológica Nacional, Córdoba, Argentina. Jornadas Argentinas de Robótica.
- [10] Benjamin C. Kuo. *Sistemas de Control Automático*. Prentice-Hall Hispanoamericana S.A., séptima edición, 1996.
- [11] Conor Lynch, Michael J. OMahony, and Ted Scully. Simplified method to derive the Kalman Filter covariance matrices to predict wind speeds from a NWP model. *Elsevier Ltd.*, 62:676–685, 2004.
- [12] Jose Luis Rodríguez Magdaleno. *Trabajo Fin de Grado: Integración de sistemas electrónicos de comunicación, sensado y control remoto en vehículos aéreos no tripulados*. Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación, Universidad de Cantabria, Octubre 2013.
- [13] K. Ogata. *Ingeniería de Control Moderna*. Pearson, quinta edición, 2010.
- [14] Dr. Radhakant Padhi. *Conversion Between State Space and Transfer Function Representations in Linear Systems – I,II*. Dept. of Aerospace Engineering Indian Institute of Science, Bangalore, 2010-2011.
- [15] Mebarki R., Cacace J., and Lippiello V. *Velocity estimation of an UAV in a GPS-denied environment with IMU and visual data*, 2013.
- [16] Daniel Rodríguez Ramírez and Teodoro Alamo Cantarero. *Espacio Estados: Representación y propiedades importantes*.
- [17] Derek Rowell. *2.14 Analysis and Design of Feedback Control Systems, State-Space Representation of LTI Systems*. MIT, Massachusetts, EEUU, October 2002.
- [18] M. Sadraey and R. Colgren. A Systems Engineering Approach to the Design of Control Surfaces for UAVs. *American Institute of Aeronautics and Astronautics*, 45, 2007.

- [19] George T. Schmidt. Navigation sensors and systems in GNSS degraded and denied environments. *Chinese Journal of Aeronautics*, 28:1–10, 2015.
- [20] Dan Simon. Kalman filtering. embedded systems programming <http://academic.csuohio.edu/simond/courses/eec644/kalman.pdf>. *Internet*, pages 72–9, Junio 2001.
- [21] Csaba Szász and Zoltán Fülöp. Development of a microcontroller-based system with hardware-in-the-loop method for real-time control algorithms prototyping. *Recent Innovations in Mechatronics (RiM)*, 1, 2014.
- [22] Fernando Torres, Jorge Pomares, Pablo Gil, Santiago T. Puente, and Rafael Aracil. *Robots y Sistemas Sensoriales*. Prentice Hall, primera edición, 2002.
- [23] Merwe R. V. and Wan E. A. Sigma-point Kalman filters for integrated navigation. *Proc of the 60th Annual Meeting of the Institute of Navigation*. Dayton: Institute of Navigation, 5:641–654, 2004.
- [24] Michel van Biezen. *iLectureOnline*, <http://www.ilectureonline.com/lectures/subject/SPECIAL%20TOPICS/26>, 2016.
- [25] Elisabeth Villotas. *Formas Canónicas*. Pontificia Universidad Católica del Perú (PUCP), Perú.
- [26] Øyvind Magnussen, Morten Ottestad, and Geir Hovland. *Informe Técnico: Calibration Procedure for an Inertial Measurement Unit Using a 6-Degree-of-Freedom Hexapod*. Faculty of Engineering and Science, Department of Engineering, University of Agder, Norway.