



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Diseño e implementación de un control en bucle cerrado de bajo coste para motores de corriente continua aplicado a impresoras 3D

DOCUMENTOS

I - Memoria

II - Pliego de condiciones

III - Presupuesto

IV - Anexos

Valencia, julio de 2017

Autor: D. Eduardo M. Alonso Sánchez

Tutor: D. Ángel F. Perles Ivars

Cotutor: D. Miguel Sánchez López



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Diseño e implementación de un control en
bucle cerrado de bajo coste para motores de
corriente continua aplicado a impresoras 3D

I - MEMORIA

Valencia, julio de 2017

Autor: D. Eduardo M. Alonso Sánchez

Tutor: D. Ángel F. Perles Ivars

Cotutor: D. Miguel Sánchez López

A quienes no me vieron finalizar mis estudios

The Fremen were supreme in that quality the ancients called “spannungsbogen” — which is the self-imposed delay between desire for a thing and the act of reaching out to grasp that thing.

The Wisdom of Muad'Dib, Princess Irulan

***The Road to Dune**, (Frank Herbert, Brian Herbert and Kevin J. Anderson)*

Agradecimientos

A mi tutor Àngel, por aceptar emprender este viaje y mostrarse siempre optimista hasta el final, a pesar de que pudo tener motivos para no serlo.

A Miguel por su dedicaci3n incansable, por haber sido mi cotutor y un verdadero pilar de apoyo que ha soportado mis inesperadas visitas y preguntas.

A todos aquellos que han contribuido de una manera u otra a las distintas etapas de este proyecto, consiguiendo as3 que avanzara siempre en la direcci3n correcta; debo agradecer a Carlos S3nchez su comprensi3n de los *drivers*, a Rafael Torres sus lecciones de mec3nica, a Santiago Vidal sus nociones de estadística descriptiva, a Rub3n Puche el contestar a mis preguntas de motores sin conocerme - y a H3ctor Baviera por ponerme en contacto con 3l, a Vanesa Cuenca y Marcos Gim3nez sus explicaciones de momentos inerciales.

A quienes han formado parte del plano personal durante la realizaci3n del TFG, becarios con quien he compartido laboratorio, compaÑeros de clase encontrados durante cinco duros aÑos de aventura, y en especial, a mis padres.

Gracias a todos.

Resumen

En el presente proyecto se pretenden solucionar de forma eficaz las carencias que se encuentran en las impresoras 3D actuales, que han sido identificadas como la falta de realimentación en el sistema, y de prestaciones en sus motores.

Mediante el análisis de los distintos dispositivos que en la actualidad forman los ejes de movimiento de una impresora 3D, se han planteado y evaluado soluciones alternativas que mejoran su funcionamiento. Los elementos sugeridos se han valorado según su capacidad para resolver los objetivos planteados, y en algunos casos su rendimiento bajo condiciones de ensayo.

Los elementos escogidos representan de forma individual una mejora respecto a los dispositivos existentes. Con su montaje en una plataforma de ensayos, se logró evaluarlos en conjunto y determinar su aplicabilidad en un sistema real.

Los resultados obtenidos han permitido que la solución adoptada se postule como una alternativa real al sistema existente. Quedan, por tanto, satisfechos los objetivos de este proyecto.

Palabras clave: *motor sin escobillas, encoder, impresora 3D*

Resum

Al present projecte es pretenen solucionar de forma eficaç les mancances que es troben a les impressores 3D actuals, que han estat identificades com la falta de realimentació al sistema, i de prestacions en els seus motors.

Mitjançant l'anàlisi dels diferents dispositius que actualment formen els eixos de moviment d'una impressora 3D, s'han plantejat i avaluat solucions alternatives que milloren el seu funcionament. Els elements suggerits s'han valorat segons la seua capacitat per resoldre els objectius plantejats, i en alguns casos el seu rendiment sota condicions d'assaig.

Els elements escollits representen de forma individual una millora respecte als dispositius existents. Amb el seu muntatge en una plataforma d'assajos, es va aconseguir avaluar-los en conjunt i determinar la seua aplicabilitat en un sistema real.

Els resultats obtinguts han permès que la solució adoptada es postuli com una alternativa real al sistema existent. Queden, per tant, satisfets els objectius d'aquest projecte.

Paraules clau: *motor sense escobretes, encoder, impressora 3D*

Abstract

The present project aims to solve effectively the shortcomings found in current 3D printers, which have been identified as the lack of feedback in the system, and the lack of performance in their motors.

By analyzing the different devices that form the axis of movement of a 3D printer, alternative solutions have been considered and evaluated to improve its operation. The suggested elements have been assessed according to their ability to solve the stated objectives, and in some cases their performance under test conditions.

The chosen elements individually represent an improvement over existing devices. With its assembly in a test platform, it was possible to evaluate them together and determine their applicability in a real system.

The results obtained have allowed the adopted solution to be postulated as a real alternative to the existing system. The objectives of this project are therefore accomplished.

Keywords: *brushless motor, encoder, 3D printer*

Índice general

Capítulo	Página
Agradecimientos	VI
Resumen	XII
Índice de figuras	XV
Índice de tablas	XVII
Lista de abreviaturas	XVII
Lista de símbolos	XX
1. Introducción y antecedentes	1
1.1. Objetivo	1
1.2. Motivación y justificación del proyecto	2
1.3. Análisis de antecedentes	3
2. Estudio de necesidades, factores a considerar, limitaciones y condicio-	
 nantes	13
2.1. Delimitación del sistema a analizar y sus elementos constituyentes	13
2.2. Posicionamiento fiable	15
2.3. Movimiento uniforme	17
2.4. Coste final y ámbito de aplicación	19
3. Planteamiento de soluciones alternativas	21
3.1. Motor	21
3.1.1. Motor de corriente continua	22
3.1.2. Motor de imanes permanentes sin escobillas	23
3.1.2.1. Generalidades	23
3.1.2.2. Estructura del motor <i>brushless</i> de imanes permanentes . .	24

3.1.2.3.	Principio de funcionamiento	25
3.1.2.4.	Tipos de motores <i>brushless</i> de imanes permanentes	28
3.1.2.5.	Técnicas de control	29
3.2.	Sistema de acople mecánico	31
3.2.1.	Husillo de rosca trapezoidal	32
3.2.2.	Husillo de bolas	33
3.3.	Microcontrolador	34
3.3.1.	Microcontrolador de 8 bits ATmega2560	35
3.3.2.	Microcontrolador de 32 bits STM32F302R8	35
3.4.	<i>Driver</i> de potencia para el motor	36
3.4.1.	Implementación del <i>driver</i> a partir de semiconductores	37
3.4.2.	<i>Driver</i> integrado L6230	39
3.5.	Elemento de realimentación de la posición	40
3.5.1.	Encoders de tipo mecánico	42
3.5.2.	Encoders ópticos	44
3.5.3.	Encoders magnéticos	45
4.	Descripción detallada de la solución adoptada	49
4.1.	Motor de imanes permanentes	49
4.1.1.	Obtención de k_E y validación del motor mediante su ensayo	52
4.1.1.1.	Diseño del ensayo	53
4.1.1.2.	Puesta en marcha	54
4.1.1.3.	Resultados y datos obtenidos	55
4.1.1.4.	Obtención de k_E	60
4.2.	Encoder óptico	61
4.3.	Microcontrolador STM32F302R8	62
4.4.	<i>Driver</i> de potencia integrado L6230	63
4.5.	Plataforma de ensayos	64
5.	Conclusiones	67
5.1.	Sobre el trabajo	67
5.2.	Futuras mejoras	67
Bibliografía		69

Índice de figuras

1.1. Mendel-Prusa iteración 3 desarrollada por Josef Průša	4
1.2. Impresoras Mendel-Prusa i3 “Hephestos” desarrolladas por bq	4
1.3. Motor paso a paso NEMA 17 y distribución interior del bobinado	5
1.4. Ejemplos de uso de las correas dentadas en impresoras 3D comerciales	7
1.5. Correa y poleas dentadas de tipo GT2	7
1.6. Ejemplos de varios tensores para correas dentadas en impresoras 3D	8
1.7. Placa Arduino MEGA 2560	9
1.8. Varios <i>drivers</i> fabricados por Pololu®	11
2.1. Diagrama de bloques del sistema analizado	14
2.2. Relación de ejes de una impresora 3D cartesiana	15
2.3. Representación del movimiento trapezoidal llevado a cabo por el eje de una impresora 3D	17
3.1. Motor de corriente continua	22
3.2. Formas de onda trapezoidal y senoidal	24
3.3. Conjunto y vista de los elementos estátor y rotor de un motor <i>brushless</i>	25
3.4. Circuito equivalente de un motor <i>brushless</i>	26
3.5. Tensión de pico entre líneas	27
3.6. Combinaciones de conmutación y sentido de la corriente	30
3.7. Tensiones senoidales de alimentación	30
3.8. Ejemplo y perfil de un husillo trapezoidal	32
3.9. Representación y perfil de un husillo de bolas	34
3.10. Circuito básico en configuración bootstrap	38
3.11. Encapsulado del integrado L6230 y ejemplo de uso	40
3.12. Esquema interno del integrado L6230	41
3.13. Esquema de conexión típico para el integrado L6230	41
3.14. Ejemplo y vista interior de un encoder mecánico	43
3.15. Señal eléctrica de un encoder en cuadratura	43
3.16. Disco óptico ranurado y ejemplo de un encoder óptico	44

3.17. Ejemplo de un encoder óptico encapsulado para uso industrial	45
3.18. Funcionamiento y ejemplo de un encoder magnético	46
4.1. Sistema <i>gimbal</i> de estabilización con cámara	51
4.2. Ejemplo de dos motores <i>brushless</i>	52
4.3. Vista general del montaje de ensayo	54
4.4. Detalle de los acoples y el disco ranurado	54
4.5. Osciloscopio Hantek DSO-5102P	55
4.6. Tensión de pico de la onda senoidal 1	56
4.7. Tensión de pico de la onda senoidal 2	56
4.8. Tensión de pico de la onda senoidal 3	57
4.9. Tensión de pico de la onda senoidal 4	57
4.10. Conjunto de señales senoidales 1	58
4.11. Conjunto de señales senoidales 2	58
4.12. Conjunto de señales senoidales 3	59
4.13. Conjunto de señales senoidales 4	59
4.14. Captura de pantalla de los datos generados por el programa realizado en R	60
4.15. Placa de desarrollo NUCLEO-F302R8	62
4.16. Placa de expansión X-NUCLEO-IHM07M1	64
4.17. Diferentes vistas de la plataforma de ensayos	65

Índice de tablas

1.1. Características del microcontrolador ATmega2560	10
3.1. Diferencias entre motores PMSM y BLDC	28
3.2. Características del microcontrolador STM32F302R8	36
4.1. Comparativa entre motores de corriente continua y motores de imanes per- manentes	50
4.2. Características del motor TURNIGY HD-2212	52
4.3. Comparación del valor de las constantes	61
4.4. Comparativa entre las posibles implementaciones del <i>drivers</i>	63

Lista de abreviaturas

BLDC	—	Brushless direct current (motor) ((motor de) Corriente continua sin escobillas)
FCEM	—	Fuerza contraelectromotriz
FEM	—	Fuerza electromotriz
PMSM	—	Permanent magnet synchronous motor (Motor síncrono de imanes permanentes)
PWM	—	Pulse width modulation (Modulación por ancho de pulso)
SPWM	—	Sinusoidal pulse width modulation (Modulación senoidal por ancho de pulso)

Lista de símbolos

\hat{e}_{LL}	—	Tensión de pico entre líneas (V)
Hz	—	Hercio
K_E	—	Constante electromotriz ($V/\frac{rad}{s}$)
K_T	—	Constante de par ($N\cdot m/A$)
K_V	—	Constante electromotriz (rpm/V)
ω_m	—	Velocidad angular mecánica ($\frac{rad}{s}$)
rad	—	Radianes
s	—	Segundos

Capítulo 1

Introducción y antecedentes

En el primer capítulo de esta memoria se tratarán tanto el objetivo del presente proyecto como su justificación y motivación. Se encauzará de esta manera el inicio de la memoria hacia una visión más cercana a los objetivos reales que se pretenden alcanzar, junto a un análisis del sistema actual existente, sin olvidar aquello que justifica y motiva la realización de este proyecto.

1.1. Objetivo

El objetivo del presente proyecto es el desarrollo de un sistema de movimiento lineal de bajo coste. Dicho sistema debe incorporar un elemento de realimentación de la posición que permita un control y conocimiento preciso de los movimientos realizados. Asimismo el sistema deberá ser accionado mediante un motor sin escobillas de imanes permanentes. El mismo deberá ser dotado de un control electrónico en bucle cerrado. Tras desarrollar un sistema de movimiento lineal, el paso final sería la implementación de un sistema lineal con los elementos escogidos que permitiera evaluar la solución adoptada. El paso siguiente tras comprobar la idoneidad de los distintos elementos escogidos será la sustitución de uno de los ejes de una impresora 3D actual por el sistema de movimiento lineal desarrollado.

Todos los elementos deberán encontrarse dentro de una horquilla económica que permita su inclusión en impresoras 3D domésticas, otro de los objetivos que se desean cumplir, ya que la mejora deberá ser implementada por usuarios finales. Con el fin de lograr la consecución de los anteriores objetivos se desarrollan los puntos que figuran a continuación:

- Análisis de los distintos elementos que componen el sistema principal y de su funcionamiento, ofreciendo una visión global del mismo.
- Planteamiento de soluciones alternativas que permitan escoger la solución más

idónea, evaluando el cumplimiento de los objetivos anteriormente expuestos

- Ensayo de varios motores del tipo elegido con el fin de comprobar su idoneidad para su uso en el presente proyecto.
- Implementación de un sistema de movimiento lineal de pruebas, que permita una evaluación preliminar de las distintas soluciones alternativas adoptadas.

1.2. Motivación y justificación del proyecto

La motivación que ha llevado a la realización de este proyecto se encuentra repartida entre el ámbito personal y el académico, ambos respaldados por cambios actuales en la industria que justifican el planteamiento y realización del presente proyecto.

Desde el ámbito académico, cabe remarcar la importancia del proceso de realización de este proyecto como una oportunidad para aplicar los conocimientos obtenidos durante el grado a la resolución y análisis de un problema práctico concreto, y no solamente como el requisito indispensable para la conclusión de los estudios de grado. A su vez, representa una buena oportunidad para mejorar un sistema existente, analizándolo en profundidad y planteando alternativas que culminen en su mejora. Sin olvidar en ningún momento que se trata de afrontar por vez primera la redacción completa de un proyecto perteneciente a un caso real, tanto de análisis como de aplicación práctica.

La parte correspondiente a la motivación personal de este proyecto puedo decir que se gestó con la construcción de una impresora 3D autoconstruida de tipo Mendel-Prusa i2 en el año 2013, tras finalizar el primer año de grado. Fue el momento en que el movimiento denominado RepRap (del inglés *Replicating Rapid Prototyping Machines* - Máquinas de prototipado rápido autorreplicantes) se encontraba en auge. Dicho movimiento se inició en el año 2004 cuando apareció la idea de una máquina capaz de fabricar piezas que permitieran construir más máquinas idénticas, reduciendo los costes de fabricación de las máquinas y permitiendo una “apertura” o “liberación” de los centros de fabricación, llevándolos a la gente. Como encontramos en la propia wiki del movimiento [1], eso permitiría que usuarios domésticos y pequeñas empresas dispusieran de una capacidad de fabricación hasta ahora impensable. De la misma forma se consigue que pequeñas comunidades en países subdesarrollados tengan acceso a una tecnología de fabricación que podría cambiar su estilo de vida. Desde el momento en que construí mi primera impresora 3D quedé fascinado por el mundo de la impresión 3D y las posibilidades que ofrecía. A partir de entonces, supe que quería modificar o contribuir a la modificación de algún aspecto relacionado con las impresoras 3D que condujera a una mejora de las mismas. Una oportunidad inmejorable para poder emprender un proyecto de estas características

era la realización del Trabajo Final de Grado.

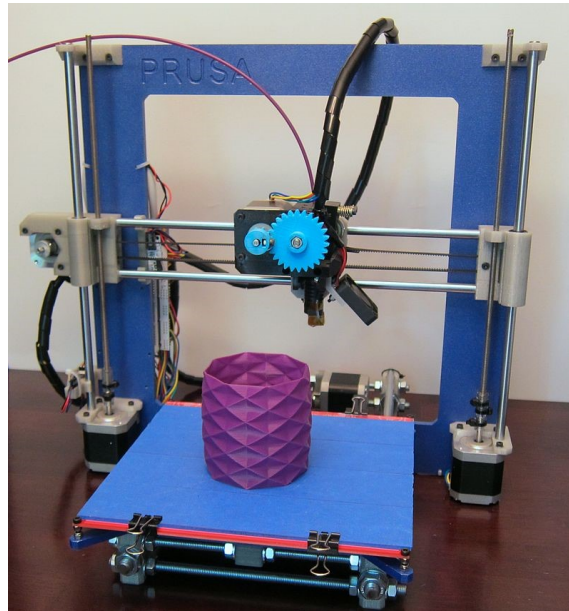
En el ámbito industrial y científico encontramos un interés creciente por los motores sin escobillas. Su reciente expansión puede verse en el ámbito de la robótica industrial y en el de la fabricación de electrónica de consumo. Los motores de corriente continua con escobillas usados típicamente en impresoras 2D de inyección y láser están siendo reemplazados por motores sin escobillas, cuyo desgaste es insignificante, y sus prestaciones mucho mayores. En el ámbito industrial, los brazos robots cuentan ya con motorizaciones basadas en motores sin escobillas, lo que garantiza una gran fiabilidad al carecer éstos de un elemento que se encuentre sometido a un desgaste permanente como serían las escobillas. De la misma manera, las prestaciones que puede ofrecer un motor sin escobillas, y por encima de todo la elevada relación par/peso del motor, los convierten en una opción inmejorable para su implementación en la robótica industrial. Otros avances que han conducido a un aumento del uso de motores sin escobillas ha sido la mejora de los imanes permanentes y la mejora de las electrónicas de control necesarias para conmutarlos.

Una mezcla de todo lo anteriormente expuesto, junto a un cruce de posibilidades y oportunidades, han sido las que han conducido a la elección y realización de este trabajo final de grado.

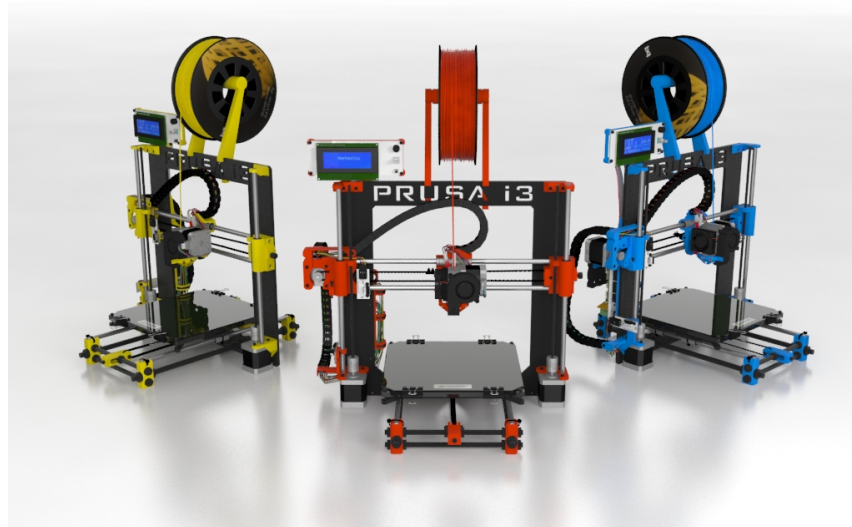
1.3. Análisis de antecedentes

Si se analiza el sistema de movimiento de las impresoras 3D autoconstruidas actuales, podemos observar que sus elementos no han sufrido mejoras en varios años, especialmente en el modelo de impresora más extendido, la Mendel-Prusa i3. Josef Průša creó este modelo en el año 2012, basándose en la Mendel-Prusa i2, cambiando el nombre referente a la iteración por *i3* tras su diseño y publicación. La imagen 1.1 muestra un ejemplo de la impresora original desarrollada por Průša en el 2012. Representada en la figura 1.2, podemos ver un modelo de impresora Prusa i3 comercializado por la empresa española bq. Los subsistemas de esta impresora no han sufrido una mejora que implicara un avance significativo, a pesar de las carencias que en determinados momentos resultan claramente manifiestas en el mismo. No obstante, si han surgido versiones, como la anteriormente indicada desarrollada por bq, que incluyen algunas mejoras que han surgido de la comunidad de impresores 3D.

Para establecer los antecedentes del proyecto y determinar el estado actual de la cuestión, se analizará el sistema de movimiento lineal del que hacen uso los ejes de las impresoras 3D actuales, examinando cada elemento de forma individual con el objetivo de describir cual es la solución actualmente usada. Además lograremos establecer una conexión entre todas los dispositivos que componen el sistema.



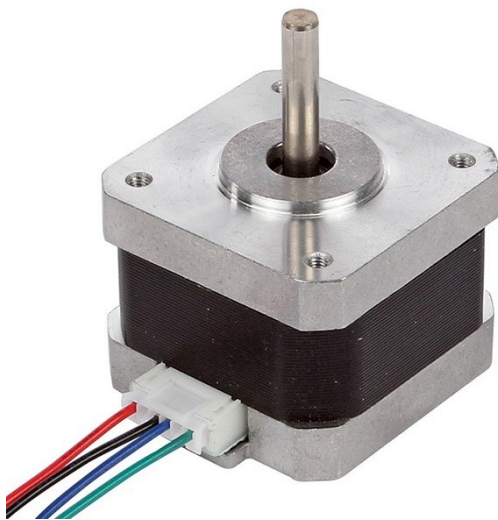
*Fig. 1.1: Mendel-Prusa iteración 3 desarrollada por Josef Průša
Fuente: botbuilder.net*



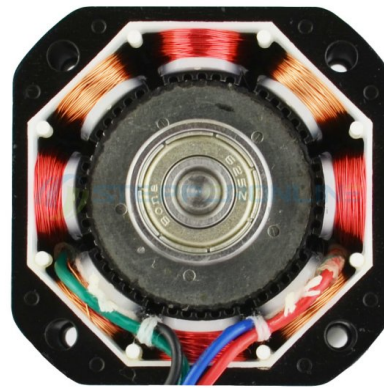
*Fig. 1.2: Impresoras Mendel-Prusa i3 “Hephestos” desarrolladas por bq
Fuente: reprap.org*

El primer elemento que se considerará dentro del sistema de movimiento lineal de las impresoras serán los motores que actúan sobre cada uno de los ejes. Estos motores son del tipo “paso a paso”, el cual es un tipo de motor de imanes permanentes que se caracteriza por su movimiento en pasos. Es un motor de imanes permanentes cuyas bobinas en el estator e imanes en el rotor se encuentran dispuestos de forma tal que los

campos de excitación provocan su movimiento a “pasos” y no en un avance continuo. Mediante la secuencia de control adecuada, se puede hacer rotar el motor un número de pasos determinado, el ángulo de los cuales varía según el bobinado y características del motor. Un ángulo de paso común en los motores empleados en las impresoras 3D es el de 1.8° por paso, por tanto se puede mantener energizado el motor de forma precisa a intervalos de 1.8° , obteniendo de esa forma una resolución de $360^\circ/1.8^\circ = 200$ pasos por revolución. En la figura 1.3(a) se puede ver el tipo de motor más empleado en las impresoras autoconstruidas, el motor paso a paso de estándar NEMA 17. En la figura 1.3(b) se ve el bobinado interior de un motor paso a paso y el rotor de imanes permanentes, el cual permite su control en pasos regulares de 1.8° .



(a) Motor paso a paso NEMA 17
Fuente: *sainsmart.com*



(b) Bobinado de un motor NEMA 17
Fuente: *electronilab.co*

Fig. 1.3: Motor paso a paso NEMA 17 y distribución interior del bobinado

Si se hace un rápido análisis de los motores paso a paso, podrá verse como estos presentan algunas deficiencias que son inherentes a los mismos. Varias de esas carencias resultan obvias durante el funcionamiento de las impresoras 3D o en el momento en que se intentan alcanzar parámetros de movimiento más avanzados, como podrían ser una aceleración o velocidad más alta. En la siguiente lista, reelaborada desde [4], se encuentran algunas de las mayores desventajas generales de los motores paso a paso:

- El par decrece rápidamente con la velocidad, ya que el par es inversamente proporcional a la velocidad en los motores paso a paso. Se traduce en la incapacidad para alcanzar velocidades altas, donde el par aplicado por el motor para mover el eje puede llegar a ser insuficiente.

- Si se produce una sobrecarga de par y el campo magnético del motor desliza, éste es incapaz de recuperarse y continuar girando.
- La relación entre el tamaño y peso del motor, y la potencia de salida disponible en el eje, es baja. Se trata de un motor pesado y voluminoso comparado con otros motores existentes.

Como puede verse, la mayoría de las desventajas se relacionan directamente con la capacidad para mover el eje con soltura, y las deficiencias que la falta de par provocaría. No obstante, lo extendido de los motores paso a paso en el mundo de las impresoras 3D se debe a una serie de ventajas como las que se enumeran en la siguiente lista, reelaborada desde [4], que los han convertido en la opción elegida en casi la totalidad de casos para el montaje de impresoras 3D:

- Bajo condiciones de operación normales, no requieren de elementos de realimentación de la posición. El propio motor es un transductor preciso de la posición.
- Par a velocidades bajas muy elevado. Útil para mover cargas a baja velocidad sin necesidad de una reductora. Resulta conveniente en las impresoras 3D al no ser indispensable realizar movimientos a alta velocidad.
- Se trata de un motor sin escobillas. La eliminación de los elementos mecánicos de conmutación entre el estátor y el rotor radican en un menor mantenimiento y mayor fiabilidad.

Los puntos anteriormente expuestos ofrecen una visión global de las carencias que pueden presentar los motores paso a paso, así como ayudar a comprender los motivos de su gran extensión dentro del mundo de las impresoras 3D que han llevado a una implementación general de los mismos.

Tras conocer el tipo de motor empleado en las impresoras 3D, tanto en las impresoras 3D comerciales como las autoconstruidas, se debe ahora conocer cual es el sistema de transmisión mecánica actualmente usado para convertir el movimiento de rotación del motor en el movimiento lineal necesario para el desplazamiento de cada uno de los ejes de la impresora. La solución adoptada para este fin es común entre las impresoras comerciales y las autoconstruidas, ambas hacen uso del mismo sistema: poleas dentadas y una correa de transmisión. Dicho sistema de transmisión de movimiento permite convertir, de forma eficiente, el movimiento rotatorio del motor en movimiento lineal, necesario para desplazar los ejes de la impresora, generalmente los ejes cartesianos X e Y. Una muestra de su aplicación se puede encontrar en la figura 1.4, donde se ve la correa dentada acoplada a un motor paso a paso en una impresora Zortrax M200 en la figura 1.4(a), y un ejemplo

en la imagen 1.4(b) de la distribución de movimiento a través de los ejes X e Y de una impresora modelo Ultimaker 3. Los elementos que componen el sistema de transmisión, poleas y correas dentadas, aparecen representados en la figura 1.5. Las poleas y correa de la imagen son del tipo GT2, que hace referencia al paso entre dientes y al perfil de los mismos. Actualmente el perfil de correa GT2 es el estándar de facto en las impresoras 3D, ya que ofrece una mejor transmisión entre la propia correa y las poleas dentadas.



(a) Motor paso a paso con polea dentada y correa

Fuente: support.zortrax.com



(b) Eje X e Y de una impresora Ultimaker 3

Fuente: maker13.com

Fig. 1.4: Ejemplos de uso de las correas dentadas en impresoras 3D comerciales

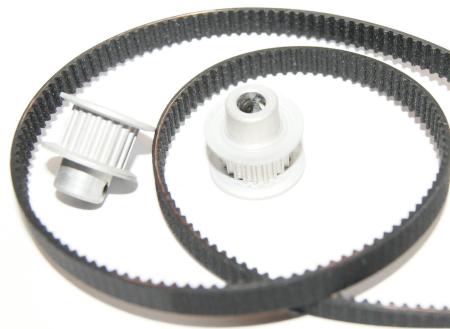


Fig. 1.5: Correa y poleas dentadas de tipo GT2

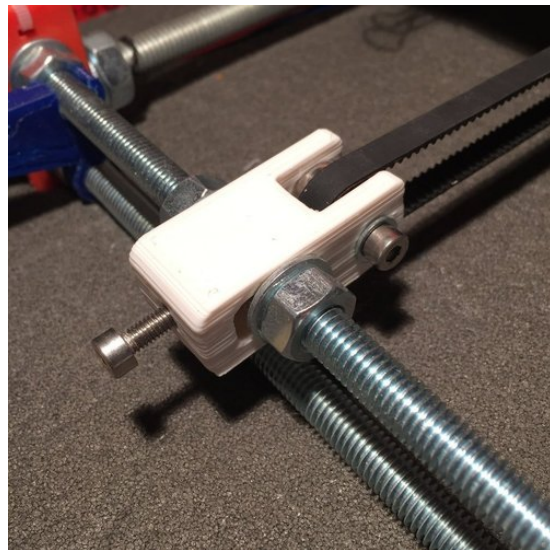
Fuente: digipak.com

Las poleas se fabrican en aluminio, que resulta fácil de mecanizar y resistente para la aplicación en la que serán usadas; las correas por su parte son fabricadas en caucho, reforzado internamente con fibra de vidrio. Tanto las correas dentadas como las propias poleas, poseen una holgura que resulta inherente a los mismos, y es debida principalmente a dos factores: la tolerancia con que correas y poleas son fabricadas, y a la elasticidad de la propia correa. Para tratar de minimizar el efecto que estas holguras pueden causar a

la hora de transmitir el movimiento, se debe mantener la correa tensada. La tensión debe resultar suficiente para asegurar un contacto firme entre los dientes de la polea y los de la correa para que no deslicen, pero no debe ser excesiva, pues podría provocar una fricción demasiado elevada en los rodamientos o en el propio motor. Mantener un nivel de tensión adecuado resulta una tarea complicada, las impresoras 3D comerciales solucionaron ese problema haciendo uso de correas cerradas, cuya dimensión es conocida con precisión, y montándolas pretensadas en la propia impresora. Debido al uso de correas abiertas, en las impresoras 3D autoconstruidas debe recurrirse a tensores u otros elementos que permitan un ajuste dinámico de la tensión de la correa. Un ejemplo de aplicación de estos tensores se puede ver en la figura 1.6, una versión simple fabricada con piezas impresas en la imagen 1.6(a) donde ha incorporado sobre la propia correa modificando su longitud; y un tensor que actúa a su vez como polea conducida en la imagen 1.6(b), permitiendo además el ajuste manual de la tensión.



(a) Tensor para correa dentada simple
Fuente: thingiverse.com



(b) Tensor para correa dentada actuando como polea conducida
Fuente: pinshape.com

Fig. 1.6: Ejemplos de varios tensores para correas dentadas en impresoras 3D

Una vez analizada la opción actual adoptada por las impresoras 3D a nivel mecánico, elemento motor y sistema de transmisión, se debe contemplar el control electrónico. Todas las impresoras 3D se encuentran controladas por un *firmware*, el cual no es más que el programa cargado en un microcontrolador que gobierna y controla todos los parámetros de cada impresora de forma autónoma. El *firmware* es el encargado del cálculo de las trayectorias de movimiento de los ejes, del control de temperatura, y de todas las variables que afectan a la impresión. Desde los inicios de las impresoras 3D autoconstruidas, se optó

por cargar el *firmware* en una placa de desarrollo abierto, que permitiera hacer de una plataforma de *hardware* libre y gratuito. La placa escogida por su número de entradas y salidas digitales, y contar con una velocidad de cálculo suficiente, fue la placa Arduino MEGA 2560, la cual puede verse en la figura 1.7.

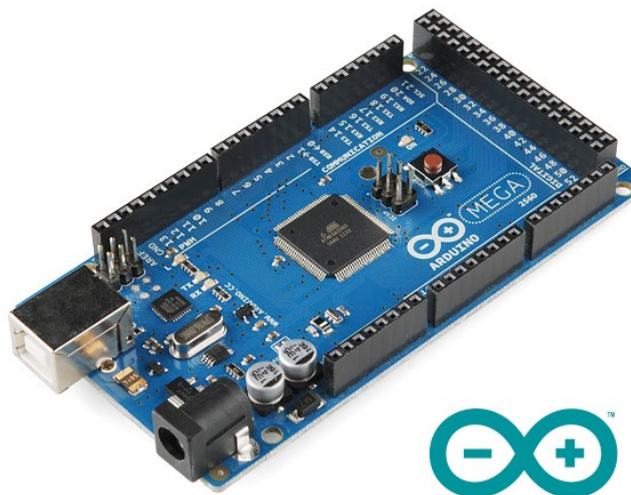


Fig. 1.7: Placa Arduino MEGA 2560

Fuente: arduino.cl

La placa seleccionada forma parte del proyecto de *hardware* abierto Arduino, el cual es un proyecto cuyo objetivo es desarrollar plataformas sobre las que programar, de forma libre y gratuita, permitiendo así a cualquier persona desarrollar sus propias aplicaciones sin restricciones y manteniendo un coste bajo. El soporte del proyecto Arduino, junto con su sencilla programación a través del entorno de Arduino y el bajo coste para el usuario final, la convirtieron en la plataforma más idónea para ser la portadora del *firmware* bajo el que se moverían las primeras impresoras 3D. La placa Arduino MEGA 2560 cuenta con numerosos conectores que permiten una fácil conexión con todos los pines disponibles en el microcontrolador como entradas y salidas. El microcontrolador montado en la placa Arduino MEGA 2560, es un ATmega2560, fabricado por la empresa Atmel Corporation®. La tabla 1.1 contiene las características principales del microcontrolador.

El microcontrolador ATmega2560, montado en las placas de desarrollo Arduino MEGA 2560, resulta suficiente para el control de una impresora 3D y de los cálculos necesarios para el movimiento de la misma. Siguiendo la estela de las impresoras autoconstruidas, que desarrollaron su electrónica de control en torno a esta placa, numerosas impresoras 3D comerciales han desarrollado sus propias soluciones electrónicas usando el mismo microcontrolador. Su memoria de programa es suficiente para contener todos los algoritmos, pero podría resultar insuficiente si se implementara un control más avanzado o se aumentara la velocidad de movimientos, provocando un aumento en la velocidad de cómputo.

Tensión de alimentación	5 V
Salidas/Entradas digitales	54 (15 con capacidad PWM)
Entradas analógicas	16
Arquitectura	8 bits
Memoria FLASH	256 KB (8 KB reservados para el <i>bootloader</i>)
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

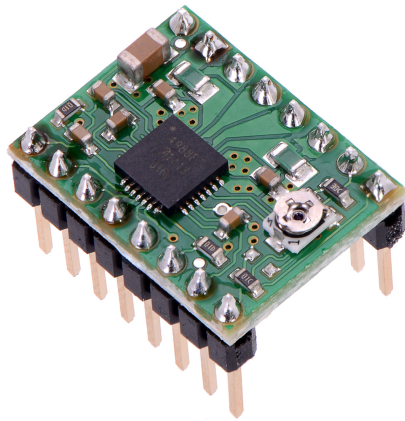
Tabla 1.1: Características del microcontrolador ATmega2560
Fuente: Elaborado a partir de arduino.cc

Debe tenerse en cuenta que la arquitectura de 8 bits del microcontrolador, junto con su frecuencia de reloj de 16 MHz lo hacen apto para la programación usada actualmente en las impresoras 3D, no obstante, las mejoras y ampliaciones de periféricos y algoritmos de control, se acercan cada vez más al límite que un ATmega2560 es capaz de alcanzar físicamente.

La electrónica de control hace uso de valores de tensión lógicos TTL, es decir, 5 V. Dichos valores de tensión a la salida del microcontrolador, junto con la limitación de corriente de salida a unos pocos miliamperios, resultan totalmente insuficientes para alimentar los motores paso a paso NEMA 17. Con el fin de mover los motores con la eficiencia requerida, se hace uso de una interfaz entre la electrónica de control y los propios motores. Para conseguir esto, se hace uso de un denominado *driver* de potencia para motores paso a paso, que resulta ser el elemento de conexión entre el microcontrolador y el motor.

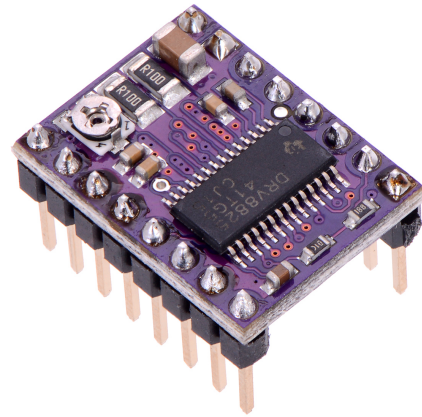
Por su sencillez para recibir instrucciones de control y sus capacidades avanzadas de control de movimientos, se optó por el uso de *drivers* de tipo integrado. Estos controladores, cuya versión más extendida es una placa fabricada por Pololu® que monta un integrado de Allegro®, pueden verse en la figura 1.8. Una versión antigua de los mismos se puede ver en la imagen 1.8(a) denominada A4988, y una versión más avanzada que permite un mayor número de micropasos y corriente, denominada DRV8825, en la figura 1.8(b).

Si se recuerda lo expuesto al hablar de motores paso a paso, se sabe que la resolución de un motor de este tipo es tan pequeña como el ángulo de su paso, cuyo valor típico es de 1.8°. Teniendo en cuenta la precisión de movimientos requeridos en los ejes de las impresoras 3D, este ángulo resulta insuficiente, incluso tras considerar la desmultiplicación mecánica



(a) Driver A4988

Fuente: pololu.com



(b) Driver DRV8825

Fuente: pololu.com

Fig. 1.8: Varios *drivers* fabricados por Pololu®

llevada a cabo por las poleas dentadas usadas para la transmisión de movimiento. Con el fin de subsanar esta falta de precisión, los integrados montados en los controladores desarrollados por Pololu® permiten el movimiento de los motores paso a paso, a través de los denominados “micropasos”. El *driver* representado en la figura 1.8(a) cuenta con una subdivisión de 16 micropasos, mientras que el modelo más avanzado de la figura 1.8(b), puede alcanzar una subdivisión de 32 micropasos. Dichas divisiones permitirían alcanzar una precisión de $1,8^\circ/16 = 0,1125^\circ$, y más recientemente con la implementación de los *drivers* DRV8825 una resolución total de $1,8^\circ/32 = 0,05625^\circ$. Estas precisiones angulares, unidas a la desmultiplicación mecánica, si llegan a resultar suficientes para el control de los ejes de la impresora. No obstante, la división de pasos en micropasos tiene un defecto inherente al propio motor paso a paso. La estructura interna del motor permite un movimiento en pasos completos, pero al dividirlos en micropasos, las corrientes que energizan al motor se vuelven mucho más complejas, pudiendo resultar en que la precisión real del motor no se corresponda con la precisión teórica angular calculada. Se debe tener en cuenta que el bobinado interno del motor y los imanes del rotor se fabrican para un número concreto de pasos, y que al hacer funcionar el motor a través de micropasos, se provoca la detención del rotor en un espacio ambiguo, donde las corrientes inducidas no han sido calculadas con precisión para mantener el motor en ese punto. Es decir, se trata de provocar la detención del motor en condiciones que no han sido contempladas en su diseño, y por lo tanto, podrían no resultar estables, ni eléctrica ni mecánicamente.

Una vez analizados de forma completa aquellos elementos que forman parte de un eje de movimiento lineal de una impresora 3D, se puede decir que se han logrado estable-

cer y comprender las relaciones existentes entre los mismos. De esta forma, ha quedado claro el objetivo común que cumplen todos los elementos como parte de un sistema único, asimismo ha quedado clara la dirección que sigue el flujo de control. Las señales de control generadas por el microcontrolador, son enviadas al *driver* de potencia para lograr mover el motor, el eje del cual se encuentra enlazado solidariamente al sistema mecánico formado por poleas y correas dentadas. Tal como se desarrollará en el capítulo 2 cuando se delimiten los elementos constituyentes del sistema, en el diagrama de la figura 2.1, quedará establecida la total conexión de los elementos; así como la función desempeñada por el nuevo elemento de realimentación de la posición, que deberá ser incluido tal como se planteó en 1.1.

Capítulo 2

Estudio de necesidades, factores a considerar, limitaciones y condicionantes

En este capítulo se desarrollaran aquellas necesidades a las que el presente proyecto pretende encontrar solución, así como detenerse en aquellos factores que delimitarán las alternativas a contemplar. De la misma manera, se valorarán las limitaciones y condicionantes que supongan un marco para las alternativas que se pueden plantear, generando una referencia para las posibles opciones que se planteen. Se establecerán asimismo limitaciones para la distinción que convertiría una solución factible en una viable.

2.1. Delimitación del sistema a analizar y sus elementos constituyentes

Debem delimitarse el sistema que se analizará y los elementos constituyentes para los cuales se plantearán soluciones alternativas. El sistema a analizar será el eje de movimiento lineal de una impresora 3D, cuyos elementos actuales han sido todos referenciados en 1.3. La representación esquemática del sistema y de cada uno de sus componentes puede verse de forma global en la figura 2.1 en forma de diagrama de bloques donde, además de figurar cada uno de los elementos constituyentes del sistema, se ve la relación que existente entre los mismos.

Cada uno de los bloques de la figura 2.1 representa uno de los elementos que forman parte del sistema actual, a excepción del bloque correspondiente a la sección 3.5 que será implementado como una mejora de calado tal y como será expuesto en dicha sección, permitiendo subsanar de esa manera uno de los fallos más flagrantes presentes en las

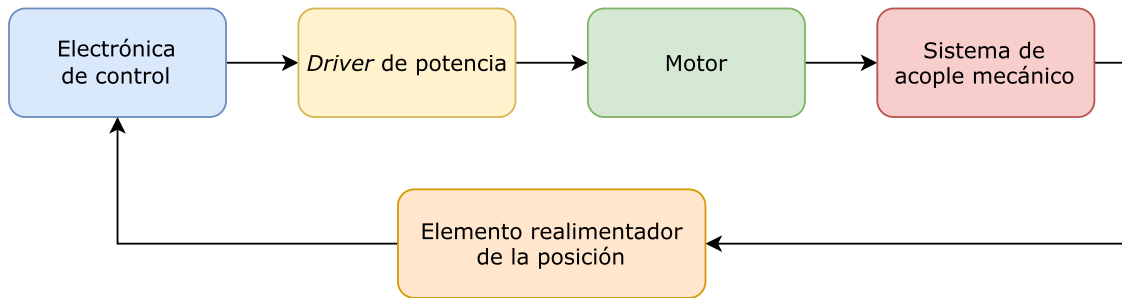


Fig. 2.1: Diagrama de bloques del sistema analizado

impresoras 3D y en su sistema de movimiento: la incapacidad de conocer de forma precisa la posición mediante un sensor de la misma, tal y como se expone en varias ocasiones en el presente proyecto. Lo que es consecuencia de un control realizado en bucle abierto, en contraposición a la implementación de un bucle cerrado planteado en este proyecto.

Tomando como base el esquema de la figura 2.1 puede verse establecida la relación de control y subordinación entre los distintos elementos, y gracias a ella se puede comprender la relación existente entre los mismos que ya se explicó en 1.3 al tratar los componentes actuales.

En la figura 2.2 se encuentran reflejados los distintos ejes que forman una impresora 3D. Al tratarse de una máquina cartesiana, basada en el conjunto de ejes XYZ, se tienen tres sistemas de movimiento lineal. Un primero correspondiente al eje X, remarcado en verde en la figura; un segundo eje coloreado en azul que corresponde al eje Y; y en tercer lugar el eje Z, marcado en rojo en la figura. El eje Z consta de dos motores y de dos sistemas lineales tal como se ha visto en las imágenes 1.1 y 1.2. Esto es debido a que se trata del eje que sustenta el peso del eje X que debe desplazarse verticalmente para conformar las distintas capas de la impresión. Para lograrlo de forma eficiente, se implementan dos motores, de esta manera se logra distribuir mejor el par necesario y estabilizar el eje al contar horizontalmente con dos puntos de apoyo.

De los tres ejes cartesianos que figuran en 2.2, se analizarán los ejes X e Y, ya que son estos dos ejes los que se encuentran formados por los elementos comunes: electrónica de control, *driver* de potencia, motores paso a paso y sistema mecánico basado en poleas y correas dentadas. El eje Z no será analizado debido a encontrarse formado por varillas rosadas a modo de husillo, resultando por tanto ligeramente más robustas mecánicamente, no obstante, todas las mejoras planteadas podrán plantearse por igual para dicho eje, ya que, a excepción de la diferencia en la transmisión mecánica, plantea las mismas carencias en el resto de aspectos.

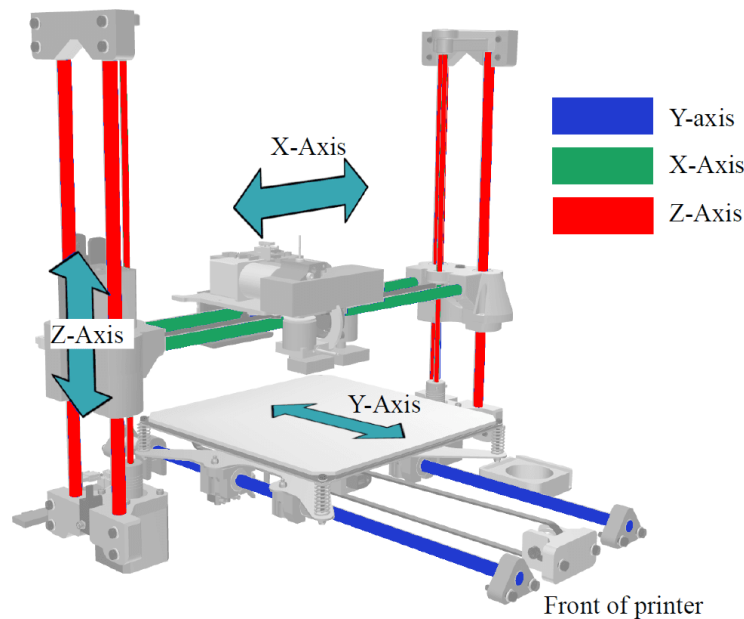


Fig. 2.2: Relación de ejes de una impresora 3D cartesiana
Fuente: mundo3dpro.es

2.2. Posicionamiento fiable

Como se ha desarrollado en el apartado 1.3 al tratar los motores paso a paso, se ha podido ver como el control sobre la posición de los motores paso a paso se basa en el principio de funcionamiento de los propios motores. Según este planteamiento, el propio motor se convierte en el transductor de la posición, de forma que los pulsos aplicados serán equivalentes al movimiento realizado por el motor sin pérdidas. Si dicha afirmación se cumpliera en todas y cada una de las aplicaciones de los motores paso a paso, no surgiría la necesidad de sustituirlos por un control que permitiera la certeza absoluta sobre la posición final.

Al tratarse de un sistema cuyo funcionamiento es en bucle abierto, sin un sistema que realmente la posición, los motores paso a paso confían ciegamente en la premisa anteriormente expuesta: la señal de control se traduce en un movimiento perfecto sin pérdidas de ningún tipo. Bajo este principio de funcionamiento perfecto, se han desarrollado todos los controles electrónicos que hacen uso de motores paso a paso para motorizar un sistema mecánico, sin contar con un sistema de realimentación de la posición.

El control actual presenta, por tanto, una carencia evidente. Pueden obtenerse pasos menores a los citados 1.8° estándar (llegando incluso a obtener 0.05625°), no obstante no hay ningún elemento de realimentación o control que pueda garantizar que dicho giro se ha llevado a cabo con la exactitud indicada por el control electrónico. Se debe recordar la no preparación de los motores paso a paso para operar con resoluciones de paso menores a

la nominal. Las causas que podrían provocar una pérdida de pasos pueden tener un origen variable y suceder en un gran número de escenarios que no pueden ser controlados por un humano. Por ello deberían ser capaces de ser controlados a través de la propia electrónica de control, que debería evaluar los movimientos realizados a través de un sistema de realimentación. La siguiente lista enumera algunas de las causas más comunes que en la actualidad provocan la pérdida de paso en los motores durante la impresión:

- Atasco mecánico en los ejes que provoque el bloqueo de los motores, siendo estos incapaces de recuperarse.
- Sobrecalentamiento de los semiconductores presentes en los *drivers* de potencia, provocando un corte transitorio de la corriente, y ocasionando una pérdida de pasos.
- Anormalidades en el propio proceso de impresión, como el levantamiento de la pieza impresa de la superficie de impresión, provocando un rozamiento excesivo con el fusor y ocasionando una breve desaceleración mecánica que cause la pérdida de pasos.

Contemplando un escenario desfavorable donde se produjera una pérdida de pasos, provocando que las señales electrónicas generadas por el microcontrolador no se tradujeran en su totalidad en desplazamiento mecánico, puede deducirse que dicho movimiento no se realizará según lo indicado. De esta manera, se estaría acarreando un error en todos y cada uno de los movimientos futuros de la máquina. No se puede asegurar la fiabilidad y precisión de los movimientos que seguirán a una pérdida de pasos.

Dentro de estas carencias, se deben contemplar también las deficiencias que presenta el propio sistema de transmisión mecánica usado, las poleas y correas flexibles presentan tolerancias y defectos en la transmisión solidaria de movimiento que pueden resultar en fallos en el desplazamiento de los ejes. Se tratará de errores de posición que no serán corregidos, y nuevamente provocarán la pérdida de precisión en los movimientos futuros, comprometiendo el resultado final de la impresión. Una pérdida de pasos pequeña, por ejemplo de unas pocas décimas de milímetro, podría tomarse por insignificante, no obstante si ese fallo se acarrea a lo largo de una pieza con múltiples movimientos se estará amplificando el error de forma considerable logrando que afecte de forma visible a la impresión en sí misma, la cual podría suponer una duración desde pocos minutos hasta decenas de horas, haciendo perder, por tanto, horas de trabajo y funcionamiento de la máquina de forma inútil.

Por todo lo aquí expuesto, se considera que la implementación de un dispositivo de realimentación de la posición es una mejora necesaria, cuyos resultados serían altamente notorios. Al incorporarse un dispositivo de realimentación en el sistema, se convierte a

la impresora 3D en una máquina que conoce su posición de forma fiable y precisa, tanta como precisión angular posea dicho elemento de realimentación.

La falta de un elemento de realimentación de la posición se considera, por tanto, un defecto en una impresora 3D debido a la multiplicidad de factores y variables que pueden provocar una pérdida de pasos que serían fácilmente evitables con la inclusión de un control en bucle cerrado.

2.3. Movimiento uniforme

Con lo expuesto en la sección 1.3 y tras conocer los distintos elementos que forman el sistema de movimiento lineal de un eje, se debe plantear el tipo de movimiento que llevan a cabo estos ejes. Como ya se ha expuesto, una impresora 3D es una máquina cuyo esquema de movimientos se basa en los ejes cartesianos XYZ. El desplazamiento a través de los mismos se lleva a cabo en tres tramos que forman un movimiento de tipo “trapezoidal”. Un primer tramo de aceleración, donde se alcanza la velocidad nominal, un segundo tramo a esa velocidad y a continuación, un tercer tramo de desaceleración para iniciar el siguiente conjunto de movimientos. Los tres movimientos necesarios, en orden de ejecución son: movimiento lineal uniformemente acelerado, movimiento lineal a velocidad constante, y movimiento lineal uniformemente desacelerado. Podría darse el caso de que el movimiento de un eje constara tan solo de dos tramos, aceleración y desaceleración. Es lo que ocurriría si no existiera distancia suficiente para que llegara a alcanzarse la velocidad nominal durante el primer tramo de aceleración, a partir del cual la máquina ejecuta un movimiento lineal no acelerado.

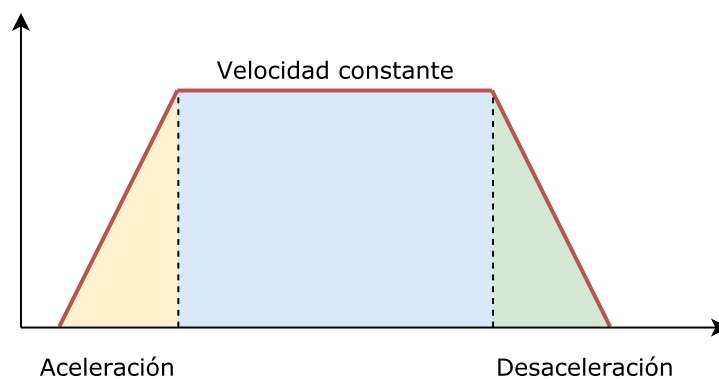


Fig. 2.3: Representación del movimiento trapezoidal llevado a cabo por el eje de una impresora 3D

Estos movimientos uniformes son calculados por el microcontrolador en función de la trayectoria provista por el archivo de coordenadas que es necesario enviar al micro-

controlador, el cual se corresponde con el modelo digital que desea imprimirse. Dichos movimientos cuentan con una aceleración y velocidades máximas que varían en función de la impresora 3D utilizada y permiten obtener un rendimiento óptimo de la impresora 3D. El *firmware* presente en el microcontrolador permite modificar estos valores, para lograr obtener el mayor rendimiento posible, que dependerá de cada máquina individualmente.

La limitación para modificar esos parámetros recae de forma principal en un elemento: el motor paso a paso. Estos motores no permiten obtener altas velocidades de giro, debido a dos factores limitantes que son:

- El incremento de velocidad provoca una pérdida de par proporcional, tal como se vió en 1.3, lo que conllevaría que a alta velocidad el motor paso a paso perdiera tanto par que provocara la incapacidad de este para mover el eje.
- Debido al propio funcionamiento del motor, una aceleración brusca o un incremento de la aceleración aplicada en un corto periodo de tiempo, provocaría el deslizamiento del campo magnético y la pérdida de sincronización entre éste y el rotor, causando el frenado del motor y del eje acoplado. Un motor paso a paso no es capaz de recuperarse de un deslizamiento magnético por si mismo. Deberá reducirse la velocidad, y volver a aplicar la secuencia de control en el orden correcto para permitir que el rotor pueda “engranar” con el campo magnético y restaurar así el normal funcionamiento del motor.

Al evaluar las inexactitudes del movimiento llevado a cabo, se deben contemplar asimismo las causadas por el sistema mecánico. El conjunto de poleas y correas de caucho dentadas resulta fiable a velocidades relativamente bajas y estables. No obstante, al pasar a velocidades más altas, se podría encontrar que las holguras presentes debido a un mal tensado o la elasticidad de la propia correa provocarían un movimiento impreciso, pudiendo llegar a ocurrir la salida de la correa dentada de la guía lateral de la polea dentada. De la misma forma, al contemplar unos cambios elevados en la aceleración, la elasticidad de la correa podría provocar nuevamente inexactitudes al absorber ésta parcialmente el movimiento realizado.

Con lo anteriormente destacado, se deberá contemplar un sistema, tanto motor como de acople mecánico, para el propio eje que permita unos movimientos uniformes estables, que además de ser estables a velocidades de funcionamiento nominales bajas, pudieran resultar igualmente fiables y exactos a velocidades altas, o frente a la aplicación de rampas de aceleración especialmente pronunciadas. Tanto el sistema motor, como el mecánico, presentan carencias y deberían contemplarse alternativas que pudieran satisfacer las necesidades planteadas.

2.4. Coste final y ámbito de aplicación

Una limitación que se debe tener en cuenta al evaluar las distintas soluciones alternativas debe ser el coste. Forma parte de los objetivos y título del proyecto que éste sea un coste bajo, al ser una mejora pensada para impresoras 3D autoconstruidas domésticas. Para resolver los problemas anteriormente expuestos y todas las carencias notables en los elementos del sistema actuales expuestos en la sección 1.3, se podrán elegir multitud de soluciones que resultarán factibles. No obstante, debido a su coste económico deberán rechazarse al tratarse de una mejora pensada para realizar en impresoras 3D autoconstruidas tal como se indica en la sección 1.1. De esta manera, el receptor final de la mejora será un usuario final, doméstico en muchos casos, que no podrá realizar un gasto económico a la altura de una pequeña o mediana empresa para mejorar su máquina.

Dentro de todas las opciones de mejora factibles analizadas, deberá tenerse en cuenta por tanto el factor económico para rechazar aquellas propuestas cuyo coste de aplicación las convirtiera en inviables para el usuario final al que van dirigidas.

De la misma manera, deberá tenerse en cuenta que se pretende sustituir un eje de movimiento lineal de una impresora 3D doméstica, y el sistema escogido deberá ser fácilmente modificable para su adaptación en las impresoras actuales, o en un desarrollo basado en las mismas. Además, deberá poder ser incorporado de una forma sencilla que permita la inclusión de las mejoras propuestas sobre el sistema existente con modificaciones mínimas.

Capítulo 3

Planteamiento de soluciones alternativas

En este capítulo se repasarán los elementos que forman parte del sistema de movimiento de un eje de una impresora 3D actual, referenciados todos ellos en 1.3, de forma que se pueda tener una visión comparativa de las distintas opciones que podrán plantearse para sustituir cada uno de los dispositivos formantes del sistema. Por cada una de las partes del sistema representadas en 2.1 se presenta una sección donde se relacionarán las diversas opciones para sustituir dichos elementos actuales que permitan lograr los objetivos establecidos para el presente proyecto. A partir de las características expuestas en este capítulo, se realizará la elección en el siguiente capítulo de aquellos dispositivos que formarán parte definitiva de la implementación del sistema.

3.1. Motor

Dentro del sistema, el motor es el único elemento actuador, siendo por tanto el único elemento a través del cual las acciones de control llevadas a cabo por la electrónica se verán reflejadas en el sistema mecánico acoplado.

Ya que éste será el único sistema que permitirá que las acciones de control se ejecuten en el sistema mecánico, deberá asegurarse, por tanto, que se escoge un elemento motriz capaz de desempeñar los movimientos con la eficacia requerida, cumpliendo los parámetros de velocidad y aceleración de forma correcta.

Tal y como se ha expuesto en la sección 1.3, el motor que se sustituirá y para el cual se buscarán alternativas, será el motor paso a paso. Dicho motor presenta unas condiciones favorables para su uso en los ejes de las impresoras 3D que han permitido su incorporación masiva en las impresoras 3D. Se deberá encontrar por tanto un punto intermedio donde otros tipos de motores puedan confluír ofreciendo de la misma manera mejoras sustanciales

que justifiquen el cambio al tipo de motor escogido.

3.1.1. Motor de corriente continua

El primer motor que se puede encontrar para sustituir el motor paso a paso del eje lineal es el motor de corriente continua. El principio de funcionamiento de estos motores es sencillo, y se basa en su alimentación a partir de una única tensión continua. Un motor de corriente continua cuenta con dos elementos básicos en los que se basa su funcionamiento: imanes permanentes en el estátor, y una bobina en el rotor. La tensión de alimentación se hace llegar a la bobina mediante unas escobillas que, mediante la fricción de las mismas con un colector de delgas dispuestas en el eje del rotor, hacen llegar la corriente hasta la bobina del motor.

La corriente se hace llegar con polaridad alterna a la bobina del rotor, lo que provoca que se generen fuerzas de atracción/repulsión de forma constante que interactúan con el campo magnético de los imanes permanentes del estátor. Gracias a esta interacción entre los campos magnéticos del motor se genera el movimiento rotatorio en el rotor. En la imagen 3.1 podemos ver un ejemplo de un motor de corriente continua de baja potencia, como los usados en impresoras 2D de inyección.

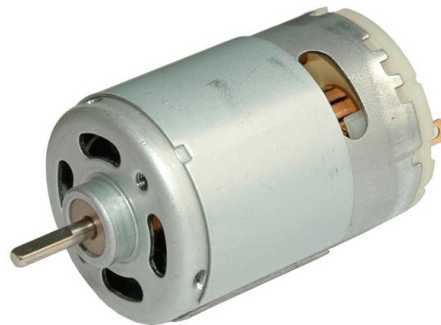


Fig. 3.1: Motor de corriente continua
Fuente: directindustry.com

Los motores de corriente continua son fiables y su control sencillo, lo que les ha hecho un hueco en la industria de la electrónica de consumo. No obstante, la relación entre el peso y las dimensiones de los mismos, y su potencia de salida, los convierten en unos motores que entregan una potencia relativamente baja en relación a su tamaño. Debe tenerse en cuenta la mayor desventaja presente en los motores de corriente continua: sus escobillas. Este elemento, que transmite por rozamiento la corriente a la bobina del rotor, sufre un desgaste mecánico y necesitará de mantenimiento y sustitución cuando su deterioro lo requiera. En el caso de un motor pequeño donde no puedan sustituirse sus escobillas, la vida útil del mismo será tan larga como la de sus escobillas, las cuales, tras un período de desgaste, provocan fallos y un menor rendimiento del motor. La fricción

entre las escobillas y el colector de delgas provoca, además, la aparición de chispas, lo que provoca un deterioro acelerado de las escobillas y la imposibilidad de usarlo en ambientes industriales peligrosos.

3.1.2. Motor de imanes permanentes sin escobillas

El segundo tipo de motor que se contempla para la sustitución de los motores paso a paso es el motor de imanes permanentes sin escobillas. Estos motores comparten una similitud con los motores paso a paso, ambos carecen de escobillas para inducir una corriente en el rotor. Cuando se habla de motores de imanes permanentes sin escobillas, se engloban todos los motores síncronos entre cuyo estátor y rotor no existe una conexión física, de aquí obtienen su característica denominación de *brushless* (del inglés “sin escobillas”).

3.1.2.1. Generalidades

Un motor de imanes permanentes se encuentra formado por tres bobinas, denominadas fases, a través de las cuales es alimentado para energizar las bobinas y producir así el movimiento rotatorio en el eje de dicho motor.

Los motores de imanes permanentes sin escobillas se pueden diferenciar por la forma de onda con la que deben ser alimentados, la cual coincide con la forma de onda de la fuerza contraelectromotriz de los mismos al funcionar como generador. La forma de onda de esta tensión puede tomar dos formas de onda que dependerán exclusivamente del bobinado del motor: senoidal o trapezoidal.

Los motores de imanes permanentes síncronos también pueden ser denominados PMSM (del inglés *permanent magnet synchronous motor*, motor síncrono de imanes permanentes), los cuales presentan una fuerza contraelectromotriz con forma de onda senoidal (figura 3.2 izquierda); por lo que se obtendrá el máximo rendimiento al alimentarlos con una señal senoidal. Estos mismos motores podrían haber sido pensados para ser alimentados no a partir de una señal senoidal, sino con un inversor a partir de una tensión continua. De ser así la forma de onda de la fuerza contraelectromotriz será trapezoidal (figura 3.2 derecha), pudiendo ser denominados motor sin escobillas de corriente continua, o bien motor BLDC (del inglés *brushless direct current (motor)*, ((motor de) corriente continua sin escobillas)).

Los motores de imanes permanentes sin escobillas tienen la gran ventaja de poseer un índice de potencia mayor que en otro tipo de motores como los de inducción, al no presentar los motores sin escobillas pérdidas en el cobre. Al no necesitar elementos de conmutación mecánica en comparación con los motores de corriente continua los hacen ser estructuras más robustas y compactas lo que, unido a la eliminación de chispas permite

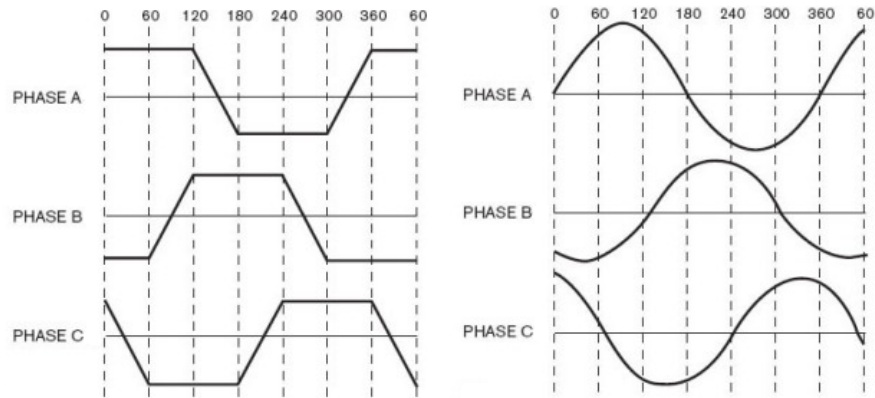


Fig. 3.2: Formas de onda trapezoidal y senoidal

Fuente: *electricbike.com*

su aplicación en ambientes intrínsecamente seguros.

Se debe tener en cuenta la aplicación para la que se están analizando los motores *brushless*. Al tratarse de una aplicación pequeña, y no industrial, los motores deberán ser dimensionados en consecuencia. Por ello, los motores escogidos como alternativa han sido los motores de imanes permanentes sin escobillas usados para la estabilización de cámaras instaladas en drones, los cuales poseen mayor par dentro de los distintos tipos de motores *brushless* junto a unas dimensiones menores a las de un motor paso a paso convencional.

3.1.2.2. Estructura del motor *brushless* de imanes permanentes

La construcción de los motores sin escobillas de corriente continua actuales es muy similar a la de los motores de corriente alterna de inducción sin escobillas. Los motores se encuentran formados por un estátor y un rotor. En el estátor de estos motores es donde se encuentran las tres bobinas que componen las tres fases de un motor *brushless*. Es posible que el número de bobinados que componen el estátor de un motor sin escobillas sea mayor que tres, pero siendo siempre un número múltiple de tres. En el rotor se encuentran los imanes permanentes, los usados en la actualidad son de neodimio, que ofrecen un campo magnético muy fuerte y resistente a la desmagnetización.

En la figura 3.3(a) puede verse un ejemplo de un motor sin escobillas, de pequeño tamaño con un peso de tan solo 39 gramos, usado para la estabilización motorizada de cámaras de vídeo incorporadas en drones. Este tipo de motores, como cualquier otro motor sin escobillas de imanes permanentes, tienen una construcción casi idéntica a la presentada en la figura 3.3(b).

En la imagen 3.3(b) se ven las dos partes constituyentes del motor separadas, mostrando los bobinados en el estátor y los imanes permanentes en el rotor. En este ejemplo

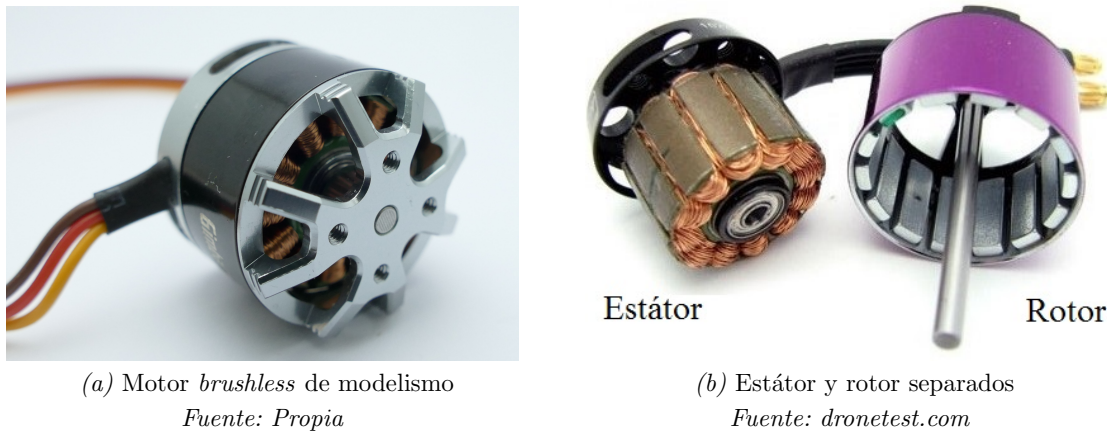


Fig. 3.3: Conjunto y vista de los elementos estátor y rotor de un motor *brushless*

concreto se pueden ver 12 bobinas en el estátor y 14 imanes permanentes en el rotor. Se comprueba que el número de bobinas es múltiplo de 3 y que el número de imanes permanentes es par, lo que garantiza la estabilidad de los campos magnéticos. El motor de la figura obtendría por tanto una denominación 12N14P, indicando la letra “N” el número de bobinados (o electroimanes) en el estátor, y la letra “P” el número de imanes permanentes en el rotor.

3.1.2.3. Principio de funcionamiento

Como se ha avanzado, el motor se encuentran formado por tres bobinados o fases, que son aquellas que se alimentan para generar un campo magnético que al ser atraído/repelido por los imanes permanentes presentes en el estátor, provocan el giro del rotor.

El estátor de un motor de imanes permanentes con escobillas puede ser modelado siguiendo el esquema 3.4, donde se pueden ver tres ramas idénticas representando una bobina o fase cada una de ellas. Dada la naturaleza de los bobinados, a partir de arrollamientos de cobre, debe considerarse el factor inductivo y resistivo de los mismos. De esta manera, se tiene que cada una de las fases se encuentra compuesta por una inductancia, L , una resistencia inherente al bobinado de cobre, R , y una fuente de tensión, e .

La conexión entre las tres fases se realiza siguiendo el esquema de la figura 3.4, de forma que cada una de las tres fases presenta uno de sus extremos sin conexión, estando éste libre y accesible en el exterior del motor para su conexión al *driver*. Los extremos opuestos de cada fase se encuentran unidos entre si formando un punto en común no accesible. Quedan de esta forma conectadas las tres fases en “estrella”, tal y como se denomina comúnmente a esta configuración.

Como ya se ha indicado, es el campo magnético generado por estas tres fases el que provoca, mediante su interacción con el campo magnético de los imanes permanentes,

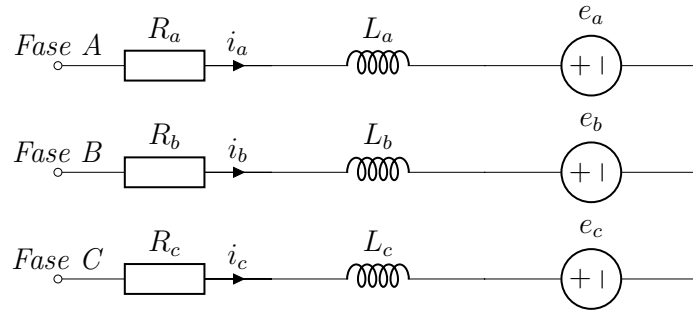


Fig. 3.4: Circuito equivalente de un motor *brushless*

el giro del rotor. Las tres fases alimentadas en la secuencia correcta provocarán una variación del campo magnético tal que permita una sincronización con el campo de los imanes permanentes y que estos giren sincronizados con la variación del campo de las bobinas.

Los motores *brushless* poseen dos constantes inherentes a los mismos que rigen su funcionamiento, la constante de par k_T , y la constante electromotriz k_E . Ambas constantes resultan definitorias para cada motor, k_T establece la relación entre la intensidad consumida por el motor y el par desarrollado por este; mientras que k_E establece la relación entre la tensión de alimentación del motor y su velocidad de salida.

Para poder comprender el funcionamiento del motor, se deben tener en cuenta ambas constantes, las cuales rigen de forma básica los motores de imanes permanentes permitiendo su control y accionamiento de forma precisa, pudiendo conocer la velocidad y par de salida a partir de ellas. No obstante, como podrá verse más adelante, ambas constantes son iguales y equivalentes entre si.

La constante k_E representa la relación que existe entre la tensión de alimentación del motor y la velocidad de rotación del rotor, tanto si el motor funciona como receptor o como generador. Las ecuaciones que rigen esta constante dependen de la distribución del bobinado. A continuación se presentan las correspondientes a los motores senoidales, cuya principal diferencia es la aparición del factor $\sqrt{3}$ en la relación entre la constante electromotriz y la constante de par.

Las siguientes relaciones de ecuaciones han sido desarrolladas desde [3] capítulo 7. La primera de ellas, muestra la expresión matemática de esta relación entre tensión y velocidad.

$$k_E = \frac{\hat{e}_{LL}}{\omega_m} \quad (3.1)$$

Donde \hat{e}_{LL} es la tensión de pico entre líneas en voltios, y ω_m la velocidad de rotación mecánica en rad/s. En el esquema 3.5 puede encontrarse la referencia para obtener la tensión

de pico entre líneas, que deberá medirse con el motor funcionando como generador y entre dos de sus fases. Las unidades de k_E son, por tanto, $V \cdot s / rad$.

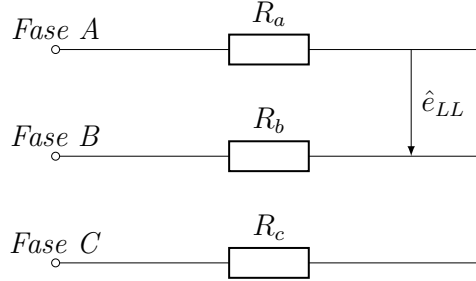


Fig. 3.5: Tensión de pico entre líneas

La constante k_T se define como:

$$k_T = \frac{T}{\hat{i}} \quad (3.2)$$

Donde T es el par expresado en $N \cdot m$, y \hat{i} la corriente de pico de línea en operación normal, expresada en amperios. Las unidades resultantes para k_T son por tanto $N \cdot m / A$.

Al tratarse de un motor senoidal, cuyos bobinados son de tipo distribuido, la relación existente entre ambas constantes se encuentra modificada por $\sqrt{3}/2$, quedando:

$$k_T = \frac{\sqrt{3}}{2} \cdot k_E \quad (3.3)$$

No obstante, si se tratase de un motor trapezoidal con un bobinado continuo, ambas constantes serían equivalentes. Para comprobar la correlación existente entre ambas constantes se desarrollarán a continuación las unidades de cada una de ellas.

Las unidades de k_T , expandidas al sistema internacional son:

$$\frac{N \cdot m}{A} = \frac{kg \cdot m^2}{A \cdot s^3} \quad (3.4)$$

Se desarrolla a continuación la definición de voltio según sus unidades en el sistema internacional:

$$V = \frac{kg \cdot m^2}{A \cdot s^3} \quad (3.5)$$

Si se divide ahora la expansión de la definición de voltio, entre rad/s ; obteniendo así la definición de k_E :

$$\frac{V}{rad/s} = \frac{kg \cdot m^2}{A \cdot s^2} = \frac{N \cdot m}{A} \quad (3.6)$$

Cuyo resultado coincide con las unidades de la constante de par k_T , demostrando así que para un motor cuyo bobinado sea de tipo continuo ambas constantes son iguales. Habiendo de relacionarse ambas entre ellas con el valor constante $\sqrt{3}/2$ debido a la distribución senoidal del bobinado.

3.1.2.4. Tipos de motores *brushless* de imanes permanentes

Dentro de la denominación de motores de imanes permanentes, se pueden encontrar los motores PMSM y BLDC tal como se ha indicado anteriormente. Ambas formas constructivas difieren en la manera en que las bobinas son arrolladas en torno al núcleo ferromagnético. Como consecuencia, sus características de funcionamiento y propiedades son también diferentes, derivadas todas ellas de los efectos que la forma de onda de la señal de alimentación produce en las bobinas.

En la tabla 3.1 se han resumido las diferencias principales existentes entre ambos tipos de motores. De esta forma resulta sencilla su comparación, tanto funcional y constructiva, como de sus características y ámbito de aplicación.

PMSM	BLDC
FEM senoidal	FEM trapezoidal
Requieren imanes más desarrollados, con mejores propiedades magnéticas y mayor resistencia frente a la desmagnetización. Se traduce en un mayor rendimiento con el mismo tamaño y peso del motor	Pueden emplearse imanes más comunes para la construcción del motor
Usos más precisos. Requieren un control electrónico para producir una señal sinusoidal	Usos más generales y de baja precisión. Control sencillo, pensado para alimentarse desde corriente continua troceada
Menor rizado de par y menor par de reluctancia, debido al menor ruido ofrece un control más sencillo	Mayor rizado de par y mayor par de reluctancia, se transmite en un mayor ruido que dificulta el control
Bobinado de tipo distribuido	Bobinado de tipo concentrado
Mayores pérdidas en el cobre	Menores pérdidas en el cobre
Alto coste de producción	Costes de producción más bajos

Tabla 3.1: Diferencias entre motores PMSM y BLDC

Fuente: Elaborado a partir de [8] p. 64

La diferenciación entre ambos tipos de motor es clara. Además de la forma de onda,

resulta evidente como los motores sinusoidales presentan un mayor rendimiento con un mayor coste, precisión y uso en aplicaciones más precisas; mientras que los motores trapecoidales resultan económicos, sencillos de controlar y su uso orientado a aplicaciones generales.

3.1.2.5. Técnicas de control

Como se avanzó al presentar los motores de imanes permanentes sin escobillas, un motor de imanes permanentes debe ser alimentado con una tensión cuya forma de onda se corresponda con la forma de onda de la fuerza electromotriz de dicho motor. Para ello, lo primero es conocer la forma de onda de dicho motor, no obstante, a pesar de la forma de onda propia, todos los motores pueden ser alimentados mediante tres técnicas básicas de control. La señal de control aplicada a través del controlador será la que determine la velocidad y el par de salida, en el momento en que las bobinas del motor “integren” la tensión de entrada en corriente.

A continuación se repasarán en los siguientes puntos cada una de las técnicas, remarcando sus características básicas de conmutación. Para ello, se ha consultado el artículo [6] donde se puede ver un repaso general y propiedades de cada técnica de control, el cual ha sido la fuente utilizada para redactar y elaborar los siguientes puntos.

Conmutación trapezoidal (*6 step*)

La primera técnica de control es la más sencilla. En este método de control se alimentan dos de las fases del motor, manteniendo la tercera de ellas desconectada y sin corriente.

De ese modo, se alterna el disparo de los MOSFET que componen ambos semipuentes para que la corriente circule a través de las seis posibles combinaciones de bobinas. En la figura 3.6 se ve representado un motor *brushless* en cada uno de los 6 posibles estados o combinaciones en los que la corriente puede fluir a través de los bobinados. El número de combinaciones posible da a este tipo de control su nombre en inglés de “*6 step*”.

Con el fin de conmutar de forma precisa cada una de las combinaciones en el momento exacto, se debe conocer la posición del rotor respecto al motor. Se tiene la ventaja de que al poseer tan solo 6 posibles combinaciones, la resolución del sistema de posicionamiento no debe ser extremadamente precisa. Una solución general a este problema pasa por usar 6 sensores de efecto Hall o similares para conocer cada una de las 6 posiciones en que debe llevarse a cabo la conmutación. Resulta una técnica de conmutación sencilla y de bajo coste, que a su vez presenta un alto rizado de par.

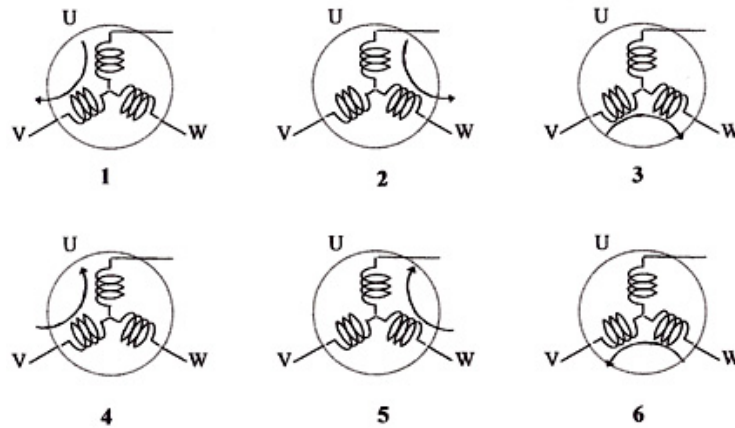


Fig. 3.6: Combinaciones de conmutación y sentido de la corriente
 Fuente: *embedded.com*

Conmutación sinusoidal

La conmutación sinusoidal representa una mejora respecto a la trapezoidal ya que ésta trata de controlar la tensión de forma continua, y no solamente en 6 intervalos por revolución.

Para ello se aplican tres tensiones sinusoidales desfasadas 120° entre ellas, representadas en la figura 3.7, a los tres bobinados del motor. El vector resultante formará un ángulo de 90° con el rotor y, por tanto, el par aplicado será máximo; evitando así la aparición de rizado de par.

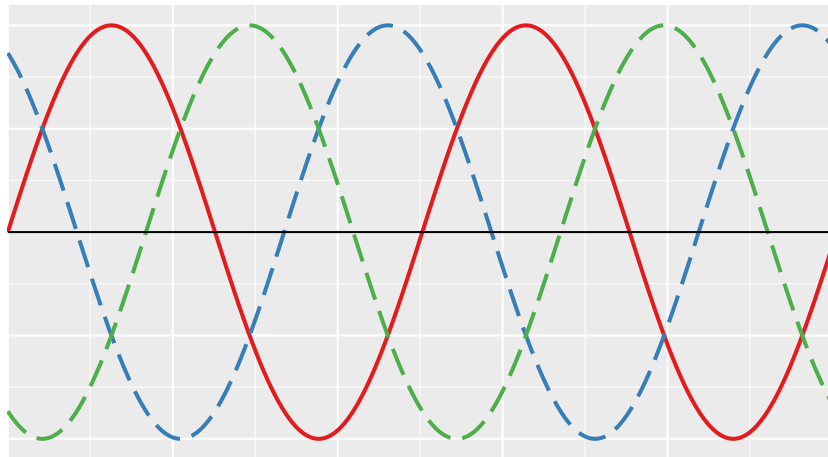


Fig. 3.7: Tensiones senoidales de alimentación

Como desventaja, el control sinusoidal debe conocer con precisión la posición del rotor para aplicar en cada instante el valor de tensión/corriente instantáneo de la onda sinusoidal correspondiente a esa posición física. Para ello, debe usarse un elemento externo de realimentación como un encoder. Es necesario, por tanto, el uso de una electrónica y un

microcontrolador capaces de calcular la secuencia de control a altas velocidades.

Control vectorial

El control vectorial es la forma de control más compleja y la que mayor potencia de cómputo necesita. De la misma manera, es también la que ofrece un mayor control sobre el motor.

Este tipo de control se basa en aplicar en cada instante temporal una señal precisa en cada una de las tres bobinas de forma que el vector corriente resultante se encuentre exactamente a 90° para provocar el máximo par posible. La alimentación de las tres bobinas de forma simultánea ya se llevaba a cabo mediante el control sinusoidal, no obstante, el control vectorial va un paso más allá al sensar exactamente la corriente que circula a través de cada bobina.

Al medir cada una de las corrientes que circulan a través de las bobinas se puede obtener un vector resultante a 90° . Para ello resulta necesario transformar las tres corrientes medidas a un espacio vectorial, calcular el vector perpendicular, y luego devolver de nuevo ese vector a un espacio basado en tres vectores. Ello se hace mediante las transformadas de Clarke y Park, las cuales permiten esta transformación de un espacio basado en tres vectores a uno basado en dos, y su inversa. En este punto radica la razón por la que es necesario un gran poder de cálculo para poder realizar todos estos cálculos con soltura y a tiempo real; sobretodo en el caso de que el motor se encuentre girando a altas revoluciones.

El control vectorial soluciona los problemas presentes en el control trapezoidal y sinusoidal, permitiendo un control estable a bajas y altas revoluciones evitando la creación de cualquier rizado de par.

3.2. Sistema de acople mecánico

Como solución alternativa al sistema mecánico actual basado en poleas y correas dentadas analizado en 1.3, se plantea la incorporación de husillos a la impresora 3D. Un husillo es un tornillo sin fin sobre el cual gira una “tuerca”, la combinación tornillo sin fin - tuerca convierte el movimiento rotatorio en un movimiento lineal. Tanto el husillo como la tuerca poseen una rosca mecanizada de precisión, la cual garantiza que la proporción entre el ángulo girado por el husillo y la distancia avanzada por la tuerca que gira sobre él, sea constante. Se trata de un elemento de transmisión cuya única finalidad es transformar el movimiento rotatorio en movimiento lineal. Dicha finalidad exclusiva hace que su mecanizado se haga con una precisión que permite su funcionamiento como elemento transmisor de movimiento de forma notablemente eficiente.

3.2.1. Husillo de rosca trapezoidal

Un husillo de rosca trapezoidal (también denominada rosca ACME), como su propio nombre indica, posee una rosca mecanizada caracterizada por un perfil de filete trapezoidal, tal como se puede ver en la figura 3.8(b). Estos husillos pueden contar con una, dos, tres o cuatro entradas de rosca, lo que afectará a la relación entre el paso y el número de vueltas completas que serán necesarias para avanzar la distancia equivalente al paso del husillo. Un ejemplo de este tipo de husillos, junto con su tuerca, puede verse en la figura 3.8. Los husillos son mecanizados típicamente a partir de barras de acero al carbono o acero inoxidable, mientras que la tuerca suele ser de latón. La variación de material entre ambos elementos evita un posible agarrotamiento por la fricción entre las dos partes móviles.

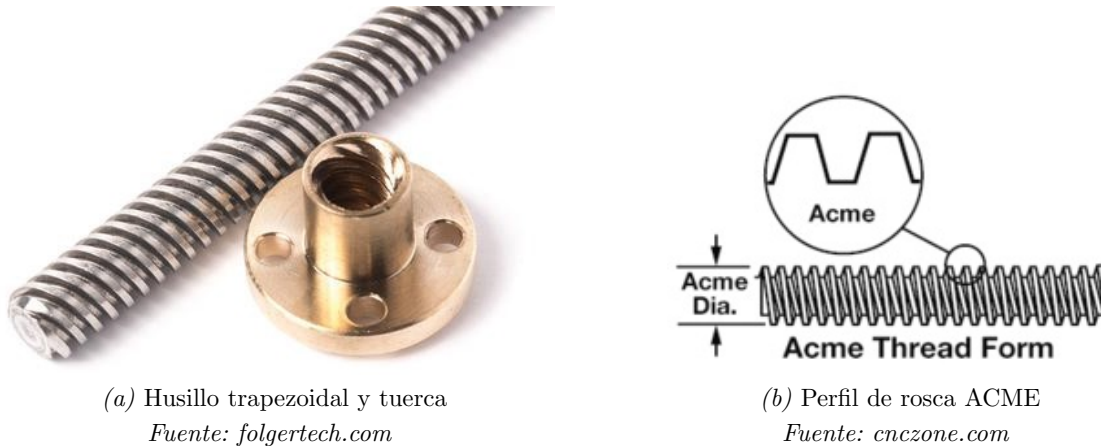


Fig. 3.8: Ejemplo y perfil de un husillo trapezoidal

Un husillo trapezoidal presenta un movimiento uniforme a lo largo de todo su recorrido, no obstante sigue poseyendo una holgura mecánica que, dependiendo del husillo escogido, podría incluso superar la de la combinación del sistema poleas dentadas - correa de caucho. Dicha holgura se produce entre los filetes del husillo y de la tuerca. A pesar de haber sido mecanizados de forma exhaustiva para llevar a cabo su cometido de la forma más precisa posible, existen unas tolerancias entre los elementos, inherentes a las máquinas que llevan a cabo el propio mecanizado, que ocasiona que haya un espacio vacío entre ambos filetes que permite el movimiento de la tuerca ligeramente en el sentido de avance sin requerir de la rotación del husillo. La aparición de este espacio adicional entre los filetes del husillo y de la rosca provoca una pérdida de precisión, ya que la relación entre ángulo girado y distancia avanzada se ve comprometida. Sería posible pues que, al avanzar el husillo un ángulo mínimo, la tuerca no avanzara.

El fallo anteriormente destacado puede ser compensado con la implementación de sencillos sistemas anti-*backslash*, como por ejemplo la inclusión de una segunda tuerca que,

mediante un muelle, mantenga una presión constante contra la tuerca original. Podemos decir que las posibles holguras son sencillamente compensables y no representan por si mismas una carencia flagrante que pudiera comprometer el sistema en su totalidad.

Los husillos trapezoidales presentan otra ventaja que se ha visto incrementada en los últimos años: su coste material y de fabricación resultan cada vez más bajos. Al tratarse de mecanizados sencillos llevados a cabo en serie sin hacer uso de las tolerancias más bajas existentes en la industria, se permite mantener el coste de estos elementos dentro de una horquilla cuyo coste mínimo y máximo son muy bajos.

3.2.2. Husillo de bolas

Un husillo de bolas (también denominado husillo de recirculación de bolas), representado en la figura 3.9(a), es un husillo mecanizado que permite transformar el movimiento rotatorio en otro lineal. Cuenta con numerosas ventajas respecto al husillo trapezoidal, las cuales se desprenden del método usado para generar el avance. A diferencia del husillo trapezoidal, el perfil del husillo a bolas que podemos ver en la figura 3.9(b) se encuentra formado por arcos de circunferencia que recorren la longitud del husillo de forma helicoidal. Estos arcos se encuentran diseñados para alojar el elemento que caracteriza a estos husillos: las bolas de acero que circulan por dentro del alojamiento que las contiene. Dichas bolas, forman el principio de funcionamiento de este tipo de husillos, la “tuerca” es un alojamiento metálico que contiene las acanaladuras necesarias para que las bolas circulen en torno a los surcos del husillo. Realmente éstas recirculan, ya que se trata de un alojamiento cerrado y las que deberían salir del alojamiento que las contiene tras finalizar su giro en torno al husillo, vuelven a la entrada del alojamiento donde comienza el giro alrededor del husillo. Una representación de las bolas girando en torno a los surcos del husillo se puede ver en la figura 3.9(a) donde las bolas se encuentran separadas por espaciadores (en color verde en la figura).

En la figura 3.9(a) se puede ver una representación 3D de las bolas de acero recorriendo los surcos del husillo, dichas bolas se encuentran separadas por espaciadores, en color verde en la figura.

Asimismo pueden verse también las salidas y entradas del alojamiento, a través de las cuales se permite la recirculación de las bolas de acero en un circuito cerrado en torno al husillo.

La transformación de movimiento rotatorio en lineal se produce gracias a la circulación de las bolas a través de los surcos semicirculares del husillo. Dichos surcos actúan como si del filete de un husillo sin fin se tratara. El filete que correspondería a la tuerca que se movería sobre él, se encuentra formado por las numerosas bolas de acero que circulan por su acanaladura. De esta forma, el giro del husillo provocará una circulación

de bolas que, debido a su bajo rozamiento, recircularán dentro de la “jaula”, representada en la figura 3.9(a). Esta recirculación de las bolas de acero a través de la tuerca y de las acanaladuras del husillo provocará el empuje necesario para generar el movimiento lineal de avance.

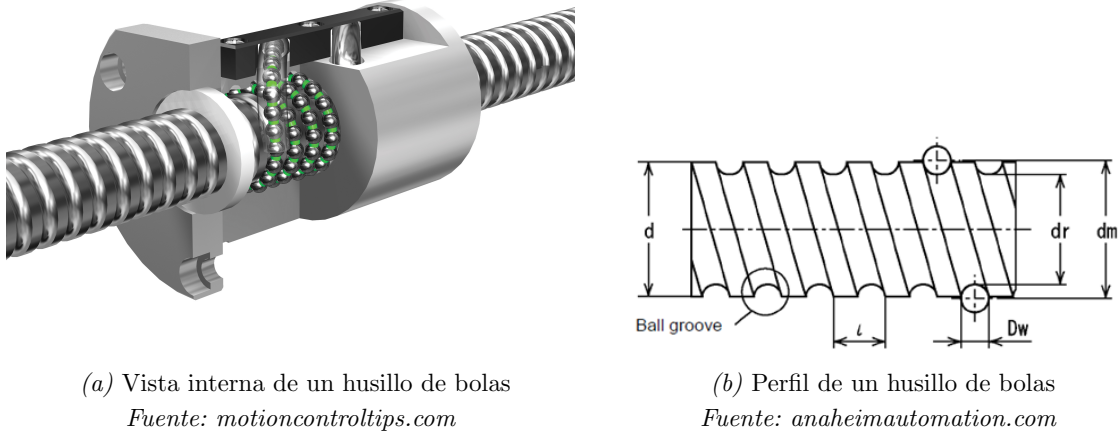


Fig. 3.9: Representación y perfil de un husillo de bolas

La ventaja que representa un husillo de bolas sobre un husillo trapezoidal común, es equiparable a la mejora que representa un rodamiento a bolas sobre un simple cojinete. La fricción se ve reducida enormemente, el par aumenta al verse distribuida la carga entre todas las bolas de acero y una mayor superficie a través de la acanaladura del husillo. Se permite una mayor precisión de movimientos gracias a la eliminación total de las holguras por la circulación de bolas de acero. El efecto de “backlash” provocado por las holguras en los husillos trapezoidales, es inexistente en los husillos de bolas.

Este elemento de movimiento lineal representa la mayor precisión alcanzable mediante elementos de transmisión mecánica. No obstante, la precisión y todas las mejoras que conllevan tienen un coste económico sustancialmente mayor que el de los husillos trapezoidales, pues requieren un mecanizado con una precisión y tolerancias mucho más ajustadas. A ello debe sumarse el coste material su fabricación en aceros de mayor calidad, y en ocasiones aceros inoxidables o cromados de alta resistencia.

3.3. Microcontrolador

Dentro de las alternativas posibles para sustituir el microcontrolador, debemos distinguir básicamente entre dos tecnologías, los microcontroladores de 8 bits y los de 32 bits. La diferencia principal entre ambas arquitecturas de construcción radica en el uso de unos registros de memoria de trabajo del procesador de 8 o 32 bits respectivamente.

A la hora de escoger un microcontrolador, se debe tener en cuenta varios parámetros que darán una referencia para poder compararlos: arquitectura interna, velocidad de reloj y número de pines de entrada/salida.

De dicho elemento, y de la programación subyacente en el mismo, dependerá de forma absoluta el buen funcionamiento de los demás elementos y de las relaciones entre los mismos. El microcontrolador a través de su programación y de un control preciso y bien ejecutado, debe ser el elemento coordinador de todo aquello que ocurre con los demás elementos, resultando por tanto su capacidad de cálculo y su buena programación elementos críticos.

3.3.1. Microcontrolador de 8 bits ATmega2560

Como primera opción se podría optar por un microcontrolador de 8 bits, es decir, el mismo tipo de arquitectura de microcontrolador que ya se usa actualmente para el control de la electrónica de las impresoras 3D.

El microcontrolador ATmega2560 expuesto en la sección 1.3, donde en la tabla 1.1 se podían ver resumidas sus características, es una opción en cuanto a la elección de un microcontrolador con esta arquitectura. Su implementación se mostraba en una placa de Arduino a modo de ejemplo en la figura 1.7.

Se trata de un microcontrolador con 54 pines configurables como entradas y salidas digitales, 15 de los cuales tienen capacidad PWM. Se debe tener en cuenta la capacidad máxima para generar una señal de salida PWM, que vendrá dada por la frecuencia máxima de reloj del microcontrolador que resulta ser de 16 MHz.

En la aplicación analizada, los microcontroladores de 8 bits se encuentran actualmente en uso controlando los motores paso a paso, tal como se vió en 1.3, proporcionando hasta ahora un resultado satisfactorio. En el caso de incluir un elemento de realimentación de la posición, el microcontrolador podría verse ralentizado a causa de la programación y a un uso intensivo de los pines de interrupción. Su coste resulta bajo, al disponer de los mismos montados en placas de desarrollo *open source* que permiten su implementación directa.

3.3.2. Microcontrolador de 32 bits STM32F302R8

La elección de un microcontrolador de 32 bits es la segunda opción a tener en cuenta a la hora de plantear la elección de un microcontrolador alternativo. En este caso, se propone el microcontrolador STM32F302R8, el cual es un integrado fabricado por STMicroelectronics perteneciente a la rama de microcontroladores STM32 con arquitectura de 32 bits basada en un núcleo ARM Cortex-M.

Este microcontrolador presenta ventajas avanzadas como la incorporación de una unidad de operaciones en coma flotante, lo que permite reducir los ciclos de reloj que son necesarios para llevar a cabo cálculos con decimales. En la tabla 3.2 pueden verse las especificaciones principales para este microcontroladores, de entre las cuales cabría destacar su velocidad de reloj junto con una mayor disponibilidad de memoria de programa en ejecución y pines con capacidad PWM.

Tensión de alimentación	3.3 V
Salidas/Entradas digitales	51 (18 con capacidad PWM)
Entradas analógicas	16
Arquitectura	ARM Cortex-M4 32 bits
Memoria FLASH	32 a 64 KB
SRAM	16 KB
Velocidad de reloj	72 MHz

Tabla 3.2: Características del microcontrolador STM32F302R8

Fuente: Elaborado a partir de la hoja de características

Los microcontroladores con arquitecturas basadas en ARM Cortex-M son un tipo de microcontroladores eficientes y aptos para todo tipo de aplicaciones embebidas. La gama M4, en la que se engloba el microcontrolador STM32F302R8, representa un equilibrio entre rendimiento y eficiencia energética. En general, se puede decir que estos microcontroladores resultan aptos para la mayoría de aplicaciones en sistemas empotrados donde se requiere un compromiso entre el consumo, la fiabilidad y el coste.

Para la aplicación analizada, un microcontrolador de 32 bits puede aportar su mayor potencia de cálculo así como un número mayor de entradas/salidas digitales, lo que permitiría una implementación más libre del sistema. La lectura del encoder que permitiría conocer la posición del sistema con exactitud se realizaría de la misma manera más rápidamente, al contar el microcontrolador con unos tiempos de interrupción menores. El coste de estos microcontroladores avanzados de 32 bits es actualmente bajo, resultando en algunos casos inferior al de microcontroladores menos potentes de 8 bits. Se dispone además de placas de desarrollo o placas de evaluación que permiten implementar un sistema de prueba con una baja inversión inicial.

3.4. *Driver* de potencia para el motor

Los *drivers* de potencia representan el elemento de unión entre el motor y las salidas del microcontrolador, dicho elemento representa, por tanto la única conexión existente

entre el control electrónico y el elemento motriz. Tal y como se ha visto en la sección 3.3, la tensión y corriente de salida de los microcontroladores resultan del todo insuficientes para controlar cualquier elemento motor, motivo por el cual se hace necesario el uso de una interfaz de potencia entre el elemento de control y el elemento motor a controlar.

Los *drivers* escogidos dependerán del tipo de motor seleccionado para sustituir a los motores paso a paso. No obstante, hay dos variantes generales que resultan típicas en cuanto a la realización de los *drivers*. Dichas variaciones en su fabricación radican en un elemento diferenciador: la inclusión o no de todos los elementos formantes del *driver* en un único circuito integrado o su montaje a partir de semiconductores y componentes electrónicos discretos individuales. A partir del tipo de motor escogido se deberá escoger en consecuencia el *driver* adecuado, sin embargo puede hacerse una elección con anterioridad sobre si se debería seleccionar una solución integrada o una ensamblada con componentes individuales. La comparación entre las distintas opciones deberá ser suficiente para que, independientemente del tipo de motor escogido, sea posible plantear el uso de una configuración de *drivers* u otra.

3.4.1. Implementación del *driver* a partir de semiconductores

La primera forma de implementación que se contemplará es aquella en la cual el *driver* se ensambla a partir de semiconductores y componentes electrónicos discretos individuales. En la figura 3.10 se encuentra un esquema típico de aplicación, donde se puede ver la configuración necesaria para el disparo de dos MOSFET. Dicho disparo se lleva a cabo mediante un diodo y dos condensadores, junto con un integrado IR2101 que permite el disparo de los MOSFET a partir del nivel de tensión presente en las salidas del microcontrolador, cosa que sin dicho integrado sería imposible. Este circuito formado en torno al integrado de control IR2101 se denomina circuito o configuración *bootstrap*. El disparo de los MOSFET se produce a través de una resistencia que se encuentra conectada a la puerta de cada uno de los MOSFET. Para controlar un motor se necesitarían una pareja de MOSFET, es decir un circuito equivalente al de la figura 3.10 por cada bobina que tenga el motor escogido. Las salidas del microcontrolador se conectarían directamente a las líneas referenciadas en el circuito de la figura 3.10 como “HIN” y “LIN”.

A través de los semiconductores de potencia principales presentes en el circuito, los MOSFET, es posible el control de una bobina tal como se ha indicado. En el caso de un motor de corriente continua sería necesaria la implementación de un solo circuito *bootstrap* mientras que si se deseara controlar un motor de imanes permanentes con tres bobinas, se debería implementar tres circuitos iguales al esquemático de la imagen 3.10.

Los MOSFETs deben dimensionarse correctamente para soportar las corrientes de arranque y las eventuales sobrecorrientes producidas por un bloqueo mecánico del motor,

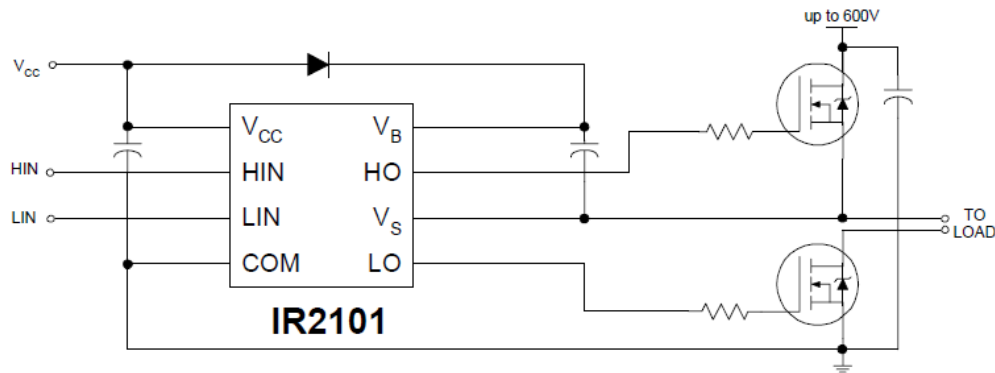


Fig. 3.10: Circuito básico en configuración bootstrap

Fuente: Hoja de características del integrado IR2101 de International Rectifier

de forma que no se sobrecalienten, o dimensionar en caso de ser necesario un adecuado sistema de disipación. Además, deberá contarse con una resistencia $R_{DS(ON)}$ lo suficientemente baja para que la potencia disipada por pérdidas en el MOSFET sea despreciable o cercana a 0.

Se debe tener en cuenta que al implementar el circuito de *bootstrap* y controlar ambos MOSFET a través de las salidas del microcontrolador, se está actuando directamente sobre el disparo de los MOSFETs. Debido a la conexión de los mismos a través de la carga, en el instante de conmutación del MOSFET de la parte superior al inferior, con el objetivo de intercambiar la polaridad de la carga, se podría provocar un cortocircuito. Los MOSFET tienen un tiempo de reacción para dejar de conducir llamado $t_{D(OFF)}$ y otro para empezar a conducir llamado $t_{D(ON)}$. En el momento de realizar la conmutación entre el MOSFET de la parte baja y el de la alta, las rampas decrecientes y crecientes de ambos tiempos coincidirían, provocando que durante un breve instante de tiempo ambos semiconductores condujeran, produciéndose un cortocircuito y la total destrucción de los semiconductores, pudiendo acarrear también graves riesgos para la alimentación usada.

Para evitar este tipo de cortocircuitos provocados por la activación simultánea de ambos MOSFET, deben incluirse *dead-times* (tiempos muertos) entre la activación de un semiconductor y el siguiente. Contemplando siempre que dicho tiempo muerto sea suficiente para permitir la total desactivación de un MOSFET antes de la activación del otro MOSFET de la misma rama.

Como conclusión para esta implementación, hay que remarcar la ventaja que representa el montaje de un *driver* a partir de componentes seleccionados de forma individual para asegurar un funcionamiento y rendimiento óptimos. No obstante, deben contemplarse dos desventajas que presenta este tipo de implementación. La primera radica en la inclusión de los citados tiempos muertos, los cuales deben ser implementados en la programación del microcontrolador y variarán según las características únicas de cada semiconductor.

La segunda está relacionada con el montaje de todos los elementos individuales que componen el circuito de la figura 3.10, los cuales deben ser ensamblados en una placa de circuito impreso diseñada *ex profeso* para este fin; el montaje resultante será voluminoso y la elección de cada componente de forma individual puede resultar tediosa unida a las pruebas de testeo. El coste final de este tipo de implementación resultará alto al tratarse de diversos componentes que deben ser adquiridos individualmente junto con el coste añadido de la fabricación de la placa de circuito impreso.

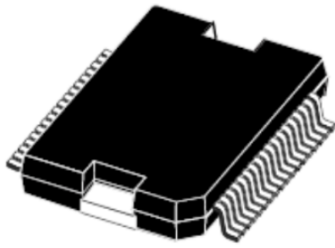
3.4.2. *Driver* integrado L6230

La segunda forma de implementación que se analizará se distingue principalmente de la anterior propuesta por incluir todos los componentes en un único integrado. El circuito que se presentaba en la figura 3.10 resulta la configuración más típica, y prácticamente la única configuración posible, para disparar de forma fiable los MOSFET desde un microcontrolador. Con esta implementación, se cuenta con todos los elementos necesarios dentro del mismo integrado. En su interior se encuentran alojados tanto los MOSFET como los elementos externos de disparo, permitiendo atacar directamente las entradas del integrado con las señales provenientes del microcontrolador y actuar sobre los MOSFET.

El chip escogido para presentar como solución alternativa es el integrado L6230, fabricado por la compañía STMicroelectronics. Se trata de un chip compacto, encapsulado en un formato de montaje superficial PowerSo36, lo que le confiere buena disipación y un tamaño reducido de la placa necesaria para incorporarlo. En la figura 3.11(a) se encuentra representado el encapsulado correspondiente al integrado L6230. En la subfigura 3.11(b) se puede ver un ejemplo de montaje, en este caso sobre una placa de expansión para la serie NUCLEO de STMicroelectronics, del integrado L6230 junto con otros componentes de montaje superficial que permiten su total control desde las salidas de un microcontrolador.

La estructura interna del integrado L6230 se encuentra representada en el esquema de la figura 3.12, donde se pueden ver todos los elementos que forman el circuito integrado y la interconexión de los mismos. De forma principal puede verse como el control del integrado gira en torno a un núcleo de puertas lógicas (*GATE LOGIC* en el diagrama). Las entradas de dicho bloque serán las señales de control, y las salidas del mismo aquellas que controlan los amplificadores que actúan sobre las puertas de los MOSFETs para dispararlos. Cada semipunto toma la corriente de la alimentación principal, siendo sus salidas las indicadas como “OUT1”, “OUT2” y “OUT3”.

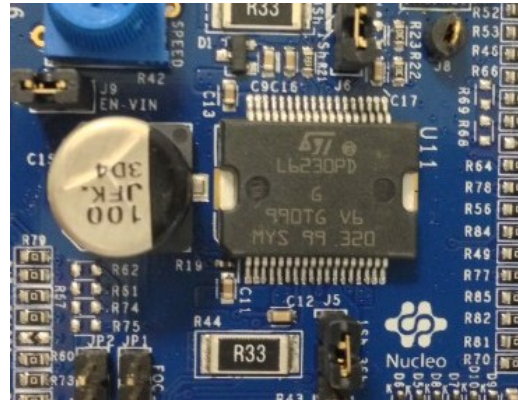
Las conexiones básicas necesarias para su funcionamiento son aquellas relacionadas con los pines de entrada del integrado que permiten su control desde un microcontrolador. Los pines marcados como “IN” y “EN”, son los pines de conexión destinados para recibir las señales provenientes del elemento de control. A través del pin “EN” se activa cada uno de



PowerSO36

(a) Encapsulado del integrado L6230

Fuente: Hoja de características



(b) Ejemplo de montaje del integrado L6230

Fuente: st.com

Fig. 3.11: Encapsulado del integrado L6230 y ejemplo de uso

los tres semipuentes de forma individual. Los pines “IN” sirven para activar el MOSFET de la parte superior del semipuerto cuando la señal se encuentra a nivel alto (un “1” lógico), y el MOSFET de la parte inferior del semipuerto cuando la señal se encuentra a nivel bajo (“0” lógico).

Por el hecho de contar con tres semipuentes en su interior, el integrado L6230 se encuentra especialmente diseñado para el control de motores con tres fases; es decir motores sin escobillas de corriente continua. No obstante, existe la posibilidad de usar el integrado para el control de motores de corriente continua con una sola fase. Como puede verse en el diagrama de la figura 3.13, el integrado L6230 necesita de pocos componentes externos discretos para que resulte posible implementar un control efectivo. Gracias a ello se consigue una implementación económica y de dimensiones reducidas.

El coste de implementación del integrado L6230 resulta bajo, al tratarse de un único integrado donde se encuentra toda la electrónica de potencia y los componentes necesarios para el control de la misma, junto con el bajo coste de los componentes discretos adicionales. Se trata de un *driver* orientado al control de un motor con tres fases que sobresale cuando se hace necesario el control de un motor de esas características.

3.5. Elemento de realimentación de la posición

Tal y como se expuso en la sección 1.3, el elemento analizado en la presente sección no se encuentra presente en las impresoras 3D actuales. Se trata de un elemento de realimentación de la posición. No obstante, éste si se encuentra incluido en el diagrama de bloques de una impresora 3D de la figura 2.1, donde además podemos ver su cometido

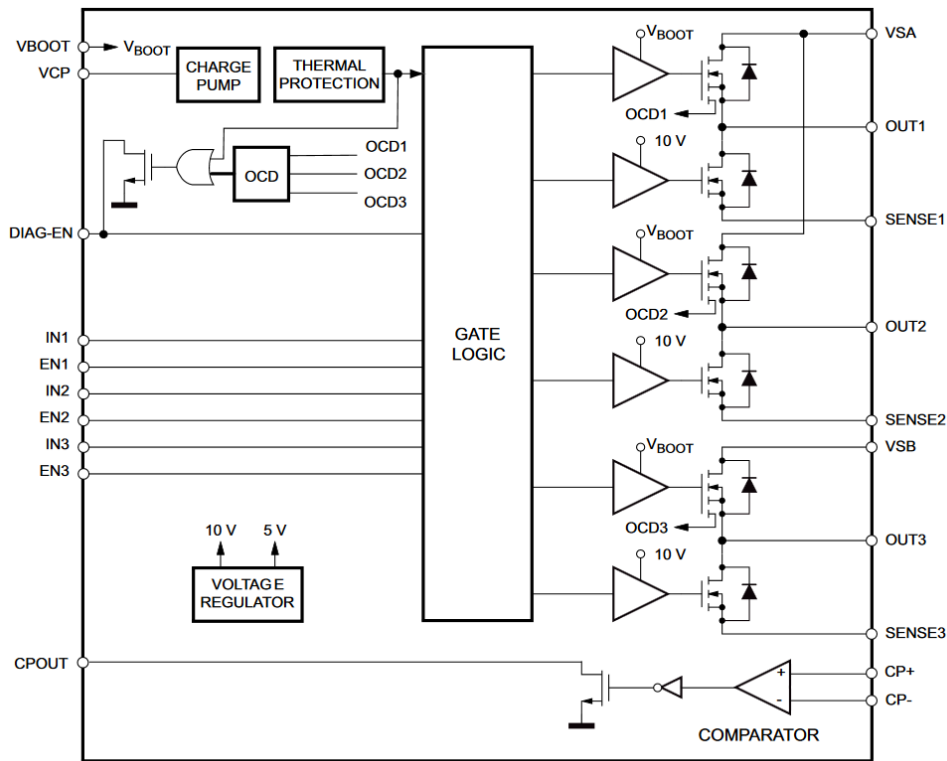


Fig. 3.12: Esquema interno del integrado L6230

Fuente: Hoja de características

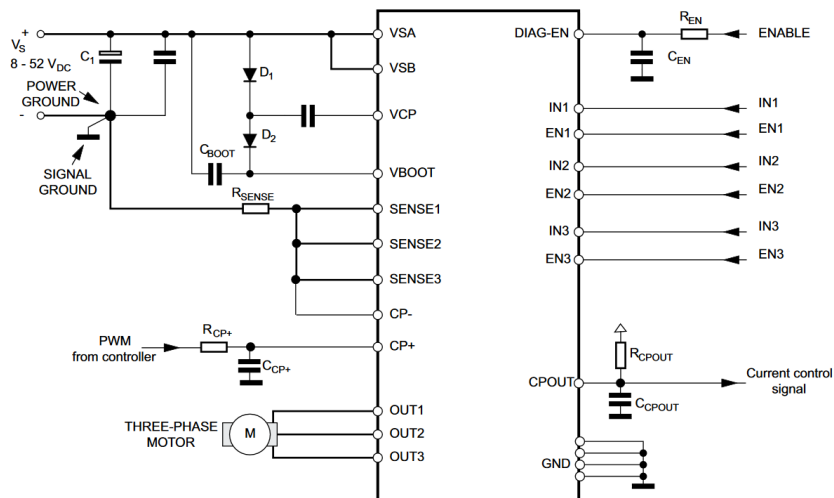


Fig. 3.13: Esquema de conexión típico para el integrado L6230

Fuente: Hoja de características

y su relación con el resto de elementos. Su función es básica, se trata de un elemento acoplado al eje mecánico de salida del motor, o bien al husillo movido por el motor, cuya señal de salida se envía al microcontrolador para que se evalúe si la posición final

deseada ha sido alcanzada con éxito. Haciendo uso de estos elementos se permite por tanto conocer la posición alcanzada realmente por el sistema físico tras llevar a cabo la acción de control por parte del motor. La precisión vendrá determinada por el propio elemento de realimentación y la precisión inherente a su fabricación.

Dentro de los elementos de realimentación se evaluarán los denominados encoders. Un encoder es un componente electromecánico cuyo objetivo es convertir la posición angular en una señal digital que pueda ser leída por un microcontrolador u otro elemento de control. Los encoders permiten que se acople a una de sus partes un eje mecánico, de forma que la posición angular medida corresponda a la posición angular de dicho eje al que se encuentra acoplado de forma solidaria. De esta forma permitirá conocer la posición angular y, por tanto, calcular medidas indirectas como la aceleración o velocidad angular, del sistema mecánico.

3.5.1. Encoders de tipo mecánico

El primer elemento de realimentación que se evaluará son los encoders mecánicos. Un ejemplo de estos encoders se puede ver en la subfigura 3.14(a). Este tipo de encoder se caracteriza por obtener la posición angular mediante un contacto mecánico entre la parte rotatoria del encoder acoplada al eje mecánico y la parte estática del encoder que se encuentra fija. Una vista interna que permite explicar su funcionamiento se encuentra en la subfigura 3.14(b), tal como puede observarse existen unas lengüetas metálicas que hacen contacto con las pistas de cobre en el interior del encoder. Dichas lenguas se encuentran en la parte rotatoria y las pistas de cobre con forma escalonada en la parte fija, de esta forma, al rotar las lengüetas que se encuentra desfasadas entre ellas sobre las pistas de cobre, se obtiene una señal eléctrica. La señal eléctrica de salida por efecto de las pistas de cobre escalonadas es de tipo cuadrada. Al tener dos señales A y B desfasadas entre ellas se obtiene una señal en cuadratura. Un ejemplo de este tipo de señales puede verse en la figura 3.15 donde se puede ver el resultado de las señales eléctricas A y B en los pines de salida y conexión del encoder visibles en la subfigura 3.14(a).

Los encoders con salida en cuadratura resultan muy útiles ya que permiten conocer el sentido de giro al producirse los flancos de subida y bajada de las señales A y B en orden alterno. De la misma manera, y aprovechando esta características, se puede subdividir cada periodo de la señal A en cuatro, gracias a la combinación con los distintos flancos de la señal B. Con esta configuración, se obtiene una resolución cuatro veces superior a la inicialmente indicada en el encoder. Se puede concluir que los encoders cuya señal de salida sea en cuadratura resultarán idóneos para el control de un eje de movimiento lineal como el de una impresora 3D. Unos ejes en los que, además de conocer la posición angular del eje, resulta crítico poder determinar el sentido de giro y, por tanto, la dirección de



(a) Ejemplo de un encoder mecánico

Fuente: amazon.com

(b) Interior de un encoder mecánico

Fuente: atariage.com

Fig. 3.14: Ejemplo y vista interior de un encoder mecánico

movimiento del eje.

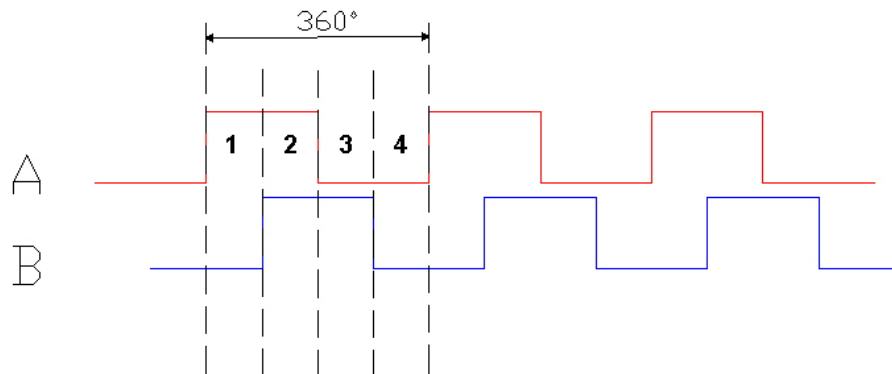


Fig. 3.15: Señal eléctrica de un encoder en cuadratura

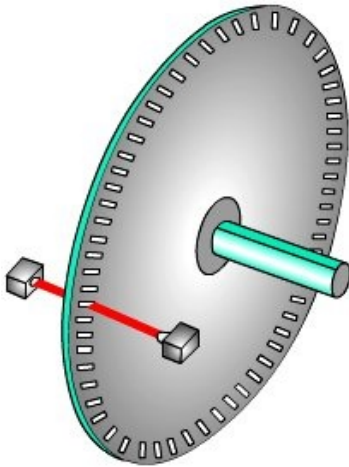
Fuente: quantumdevices.files.wordpress.com

Los encoders mecánicos son dispositivos simples que han existido en la industria desde hace años, resultan fáciles de fabricar, su sistema de funcionamiento es sencillo, basándose en el simple roce de lengüetas mecánicas con pistas de cobre. No obstante, de estos sencillos elementos se desprende la mayor desventaja que presentan estos componentes: el desgaste sufrido debido a la fricción entre las pistas y las lengüetas. Asimismo, la suciedad, el polvo o el óxido acumulado afectarán negativamente al buen contacto entre los dos elementos provocando un mal funcionamiento o incluso el atasco mecánico del encoder. La precisión de este tipo de encoders se ve limitada por el tamaño de las pistas de cobre y las lengüetas, lo que resulta generalmente en un número de pulsos relativamente bajo.

3.5.2. Encoders ópticos

Los encoders ópticos presentan el mismo modo de funcionamiento que los encoders mecánicos en cuanto a la conversión giro/señal electrónica cuadrada se refiere, pero su sistema se diferencia de forma principal en la forma de obtención de los pulsos correspondientes a cada movimiento angular. Mientras que los encoders mecánicos obtenían los pulsos a través del contacto físico entre dos elementos, en los encoders ópticos estos pulsos se obtienen sin contacto.

El funcionamiento básico se encuentra representado de forma simple en la subfigura 3.16(a), donde se puede ver un disco ranurado acoplado a un eje. Perpendicular al disco y situados sobre las ranuras, se encuentran dos elementos fotoeléctricos: un sensor y un receptor, que pueden ser de luz visible o infrarroja. El giro del disco ranurado provoca el corte del haz de luz emitido a través de las ranuras y resulta por tanto en una señal a nivel bajo del elemento receptor. Con este principio de funcionamiento, el giro del disco ranurado y el corte alterno del haz emitido, se obtiene de nuevo una señal cuadrada que se corresponde con la posición angular girada. En el caso de que el disco ranurado contara con dos pares de ranuras, se está de nuevo ante una señal de salida en cuadratura, lo que permitiría obtener de nuevo todas sus ventajas tal como se ha señalado en el apartado anterior.



(a) Disco interno de un encoder óptico

Fuente: tecnoloxia.org



(b) Ejemplo de un encoder óptico

Fuente: mazonmotor.com.au

Fig. 3.16: Disco óptico ranurado y ejemplo de un encoder óptico

Un ejemplo comercial de un encoder óptico se puede ver en la subfigura 3.16(b), donde se ve el disco ranurado con una fijación que permitirá acoplarlo a un eje, y montado sobre una placa de circuito impreso el sensor fotoeléctrico. Este ejemplo de aplicación puede

ser encontrado en impresoras láser 2D donde resulta común aplicar este tipo de encoders para conocer la posición de los distintos mecanismos internos. No obstante, tal como puede deducirse al observar la imagen, se comprueba que este montaje puede resultar no demasiado robusto para su aplicación en un sistema industrial. Por este motivo se pueden encontrar otros ejemplos de encoders ópticos como el de la figura 3.17, donde todos los elementos del mismo se encuentran encapsulados en un cuerpo metálico sellado. De esta forma, se protegen las partes delicadas del encoder, evitando uno de los mayores problemas que podrían presentarse en este tipo de encoders: la acumulación de suciedad sobre las ranuras podría evitar que el haz de luz fuera correctamente recibido por el sensor fotoeléctrico.

Con el encapsulado de la figura 3.17 pensado para un ambiente de aplicación adverso haciendo que resulte más robusto, siguen existiendo dos consideraciones en cuanto a su aplicación. La primera es la imposibilidad del eje de acople del encoder para soportar cargas axiales, lo que imposibilita su uso como contrapunto o como apoyo en el extremo de algún eje que conlleve soportar una carga. En segundo lugar cabe remarcar que este tipo de encoders deben encontrarse en un entorno mecánicamente estable, debido a que las vibraciones afectan a su buen funcionamiento interno pudiendo provocar señales falsas o incompletas. A pesar de estas limitaciones mecánicas, resultan interesantes por su alta precisión de hasta 2400 pulsos leídos en cuadratura junto a un coste de fabricación relativamente bajo.



Fig. 3.17: Ejemplo de un encoder óptico encapsulado para uso industrial

Fuente: megatron.de

3.5.3. Encoders magnéticos

Los encoders magnéticos no poseen ningún elemento de conexión entre el eje sobre el que se desea medir el giro y el propio elemento de medición. Difieren de los encoders mecánicos y ópticos en que su señal de salida no es una señal eléctrica cuadrada. Su funcionamiento se basa en el acople de un imán permanente o disco magnético al eje del

que se desea obtener la medición. El propio encoder magnético es un circuito integrado, como podemos ver en la figura 3.18(a). En su interior se encuentran una serie de sensores magnéticos que son capaces de sensar con una alta precisión la posición de un campo magnético que se encuentre sobre ellos. El disco magnético debe girar perpendicular al circuito integrado que aloja el sensor.

En la misma figura 3.18(a) se puede ver una gráfica que representa la señal de salida respecto al ángulo girado. De esta forma, a cada diferencial de ángulo girado por el eje corresponde un valor de tensión discreto. En este modo de funcionamiento, la resolución del ángulo girado será tan precisa como resolución tenga el convertidor analógico digital. Existen otros encoders magnéticos de salida digital, cuya señal de salida no es un valor de tensión continuo, sino una salida digital binaria donde la resolución del ángulo girado corresponde al número de bits de salida de dicha señal. Este tipo de encoders que cuentan con una resolución de 2^{bits} , suelen poseer la mayor de las resoluciones posibles entre las distintas tecnologías de encoders.

La imagen 3.18(b) representa un encoder magnético en un montaje típico para su uso con un eje. El disco magnético se acopla al eje, el cual pasará a través del agujero practicado en la placa de circuito impreso. Dicha placa aloja el circuito integrado que actúa de sensor magnético junto con los componentes discretos que este necesita para su correcto funcionamiento.

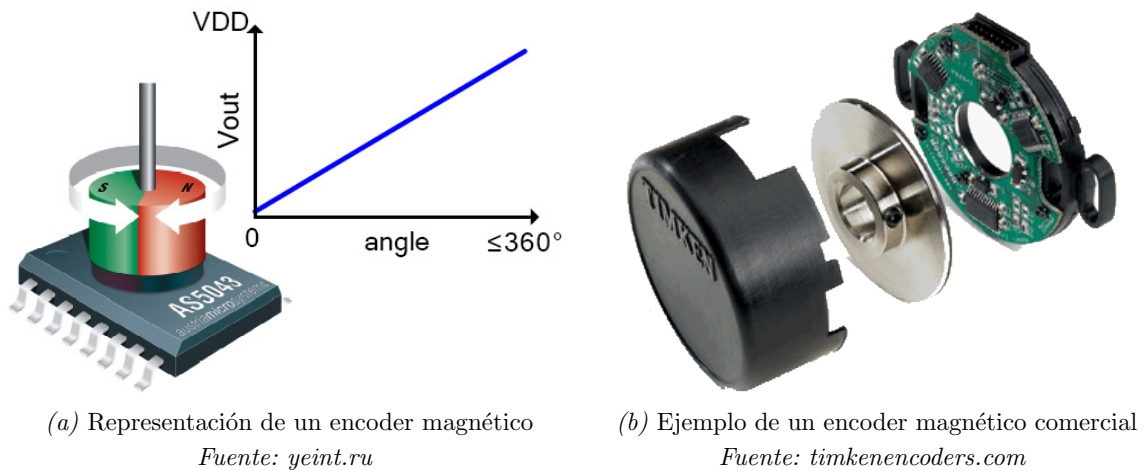


Fig. 3.18: Funcionamiento y ejemplo de un encoder magnético

Los encoders magnéticos resultan ser muy precisos, su coste de fabricación es bajo al tratarse de un circuito integrado y un disco magnético o imán permanente. No obstante, presentan una mayor dificultad al tratarse de dos elementos que deben ser instalados perpendicularmente. La implementación de este tipo de encoders presenta la desventaja que conlleva unir el elemento magnético al eje y a su vez situar el sensor magnético lo

más cerca posible pero sin existir contacto físico entre ambos elementos. Debe tenerse en cuenta la dificultad de acoplar un elemento magnético a un eje de distintos tamaños y la placa de circuito impreso necesaria para montar junto a este, lo que dependiendo del tamaño del eje podría resultar complejo. Las soluciones industriales que cuentan con ambos elementos integrados para permitir una instalación algo más sencilla resultan caros.

Capítulo 4

Descripción detallada de la solución adoptada

Este capítulo desglosará cada uno de los elementos escogidos junto con las características definitorias que han favorecido su elección frente a otras alternativas presentadas en el capítulo anterior. Se recogen de esta forma los elementos escogidos junto con un ensayo para los motores, así como la implementación inicial de los mismos en un sistema de prueba.

4.1. Motor de imanes permanentes

El motor seleccionado de forma definitiva para su estudio en la implementación de un eje de pruebas ha sido un motor de imanes permanentes sin escobillas. Como puntos clave para la elección de un motor de imanes permanentes sobre un motor de corriente continua se ha tenido en cuenta la tabla 4.1, donde se puede ver una comparativa entre ambos tipos de motores.

Las principales características que han sido consideradas para la elección de este tipo de motores, se encuentran relacionadas a continuación:

- Excelente relación entre las dimensiones y peso del motor con la potencia entregada. Correspondencia peso/potencia muy alta.
- Carencia de escobillas, lo que implica la eliminación del mantenimiento y de un desgaste que acabaría afectando al rendimiento del motor. Larga vida útil del motor.
- Resistencia a condiciones adversas de funcionamiento y posibilidad de uso en ambientes intrínsecamente seguros.

	Motor de corriente continua	Motor <i>brushless</i>
Estructura mecánica	Campos magnéticos en el estátor	Campos magnéticos en el rotor Similitud con los motores síncronos de corriente alterna
Características distintas	Control excelente y rápida respuesta	Gran durabilidad Requieren poco mantenimiento
Método de conmutación	Contacto mecánico entre las escobillas y el colector	Conmutación electrónica mediante semiconductores
Método de detección de la posición del rotor	Detección llevada a cabo por las propias escobillas y el colector	Sensor de efecto Hall, encoder óptico, etc.
Método de inversión de giro	Inversión de los terminales de conexión	Reordenación de la secuencia lógica de control

Tabla 4.1: Comparativa entre motores de corriente continua y motores de imanes permanentes

Fuente: Elaborado a partir de [8] p. 64

No se debe olvidar que el giro de la industria hacia la elección de motores sin escobillas para sus aplicaciones implica una mejora evidente de los mismos respecto a los motores de corriente continua, lo que ha significado el inicio de su implantación en robots industriales y electrónica de consumo. Se puede remarcar, por tanto, que existe una estandarización en el uso de motores *brushless* para distintas aplicaciones, tanto en ámbitos industriales como domésticos, que ha favorecido la normalización del uso de este tipo de motores para aplicaciones cotidianas.

Asimismo se debía elegir un motor cuya potencia y dimensiones se englobaran en el rango necesario para sustituir el motor que se pretendía reemplazar, el cual es el motor paso a paso descrito en 1.3. Con el fin de implementar un motor de dimensiones y prestaciones similares, se ha optado por elegir un motor *brushless* proveniente del mundo del radiocontrol. Los motores comúnmente usados en drones y otros aviones teledirigidos poseen unas altas velocidades de rotación situadas típicamente en el orden de las 30000 revoluciones por minuto, unos valores demasiado altos para esta aplicación. No obstante, encontramos otro tipo de motores que han visto un uso creciente en los últimos tiempos, los motores usados para estabilizar las estructuras que alojan las cámaras en un dron y permiten de esta forma la grabación de forma estabilizada. Este sistema de estabilización se conoce por su nombre en inglés “*gimbal*”, un ejemplo de este tipo de sistema podemos verlo en la figura 4.1. En el sistema de la imagen, tres motores sin escobillas permiten el giro y la estabilización de la cámara incorporada en todos y cada uno de los ejes posibles de movimiento.

Estos motores se caracterizan por tener unos bobinados con un reducido número de



Fig. 4.1: Sistema *gimbal* de estabilización con cámara

Fuente: *quadrone.ru*

espiras, en comparación con otros arrollamientos típicos de motores *brushless*. Esa reducción en el número de espiras conlleva un aumento de la constante de par y, por tanto, una disminución de la constante k_E . De esta manera, se obtiene un motor con un par elevado y baja velocidad, perfecto para la aplicación presentada en este proyecto; ya que los mismos están pensados para mover cámaras de pesos relativamente significativos a velocidades bajas. El mismo tipo de movimiento que realizan las impresoras 3D, movimientos lentos que requieren de un par continuo y estable.

Dentro de esta gama de motores, se tiene disponible un amplio abanico de posibilidades para elegir un motor de similares dimensiones y prestaciones. Se pueden encontrar motores cuyos diámetros varían entre los 27 mm (motor de la figura 4.2(a)) y los 63 mm (motor representando en la imagen 4.2(b)).

El motor escogido ha sido el modelo HD-2212 del fabricante TURNIGY, representado en la figura 4.2(a), cuyas características se exponen en la tabla 4.2.

Se trata de un motor cuyas características constructivas hacen que su constante k_E sea muy baja, lo que genera un par de salida elevado. Como en todos los motores cuya finalidad es la estabilización de cámaras a bordo de drones, es necesario un par elevado que permita rotar las cámaras sobre su propio eje.



(a) Motor TURNIGY HD-2212

Fuente: hobbyking.com

(b) Motor TURNIGY HD-5208

Fuente: hobbyking.com

Fig. 4.2: Ejemplo de dos motores *brushless*

Diámetro del motor	27 mm
Tensión nominal	12 V
Número de bobinados	12 (4 arrollamientos por fase)
Número de polos	14 (7 pares de polos)
Número de espiras por bobina	12 V
k_E	$0.133 \text{ V}/\frac{\text{rad}}{\text{s}}$ ($k_V = 72 \text{ rpm}/\text{V}$)
k_T	$0.133 \text{ N}\cdot\text{m}/\text{A}$

Tabla 4.2: Características del motor TURNIGY HD-2212

Fuente: Elaborado a partir de la hoja de características del motor HD-2212

4.1.1. Obtención de k_E y validación del motor mediante su ensayo

Una vez se han escogido los motores de imanes permanentes sin escobillas para la implementación de la plataforma de ensayos, deben validarse para la aplicación concreta en la que serán implementados finalmente.

Al escoger los motores de bajo coste empleados para la estabilización de *gimbals*, se está renunciando en muchas ocasiones a contar con un fabricante cuya hoja de características relativa al motor resulte completamente fiable y detallada; ya que, por lo general, los controladores son genéricos y no es necesario conocer de forma pormenorizada los detalles del motor. Por ello, resulta interesante el ensayo de los mismos con el fin de obtener al menos una de las dos constantes que rigen el funcionamiento de este tipo de motores, de modo que se pueda estimar con aceptable precisión las características del motor ensayado.

Tal y como se avanzó anteriormente, se puede ver que dentro de estos motores existen

dos constantes cuyos valores resultan definatorios y engloban en si mismas las características más relevantes de un motor *brushless*. Es por ello, que a continuación se va a plantear un ensayo que permita la obtención de la constante electromotriz k_E de forma sencilla.

4.1.1.1. Diseño del ensayo

Una vez escogido el tipo de motor, imanes permanentes sin escobillas, resulta interesante la obtención de los parámetros físicos del motor con el fin de obtener un modelo matemático del mismo para futuros controles, además de permitirnos calcular y dimensionar mejor tanto la electrónica de control como el propio sistema físico.

Para ello, se va a plantear un ensayo en el que se pueda obtener la constante k_E , de forma que será posible conocer la relación de fuerza electromotriz que rige el motor y conocer asimismo la constante de par. El ensayo planteado tratará de obtener la constante k_E de la forma más sencilla posible, para ello se seguirán los consejos encontrados en [3], cap. 7.

Como se ha visto, la relación existente entre la velocidad mecánica, ω_m , y la tensión de pico de línea generada por el motor al funcionar como generador, \hat{e}_{LL} , viene dada por la siguiente expresión.

$$k_E = \frac{\hat{e}_{LL}}{\omega_m} \quad (4.1)$$

Seguindo la ecuación anterior, se desprende que si se hace girar el motor a una velocidad conocida y se mide la tensión de pico que se genera entre dos de sus tres fases, se puede obtener k_E de forma sencilla, y lo más importante, de forma directa siempre que se opere con unidades del sistema internacional

Para llevar a cabo esas mediciones se ha montado la plataforma de ensayo de la imagen 4.3, donde se puede ver un motor de corriente continua (izquierda), acoplado a un motor de imanes permanentes (derecha).

El planteamiento resulta por tanto sencillo: se aplica una tensión al motor de corriente continua, y éste provoca que el motor de imanes permanentes gire de forma solidaria. Al girar, se puede medir en sus terminales la tensión entre fases. La velocidad de giro se mide mediante la incorporación de un sencillo disco ranurado que servirá de encoder.

En la imagen 4.4 puede verse un detalle de los acoples entre el motor de corriente continua y el motor *brushless*, así como el disco de 4 ranuras usado como encoder. El acople entre el motor de corriente continua y el del motor *brushless* ha sido realizado mediante un trozo de tubo termoretráctil el cual absorbe las vibraciones indeseadas con su comportamiento flexible. El disco ranurado y el acople mecánico del motor de imanes permanentes han sido fabricados mediante una impresora 3D.

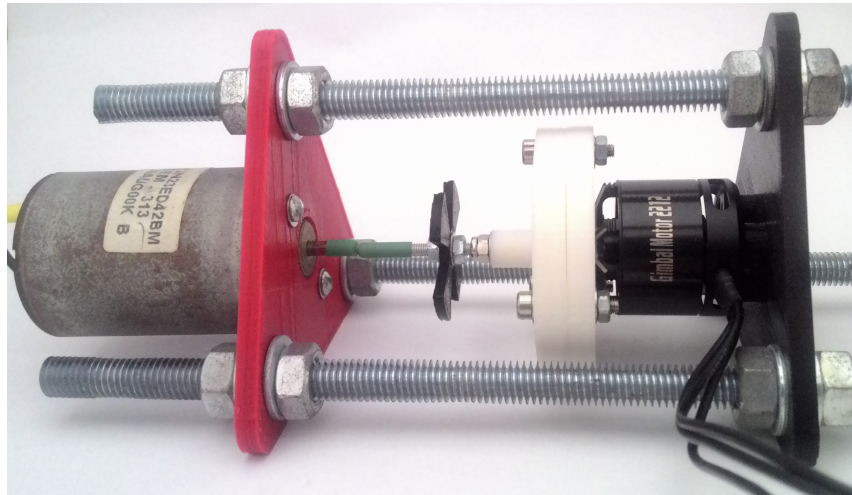


Fig. 4.3: Vista general del montaje de ensayo

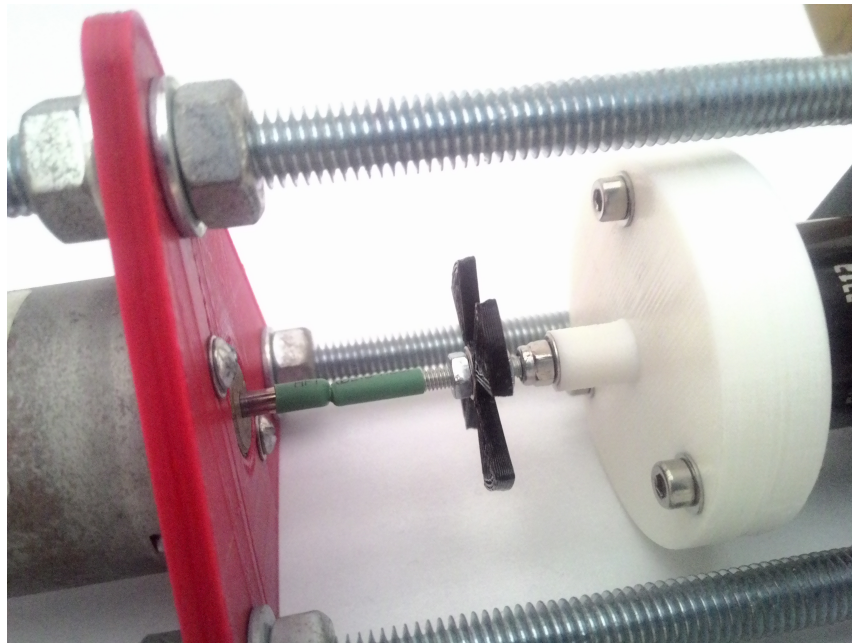


Fig. 4.4: Detalle de los acoples y el disco ranurado

4.1.1.2. Puesta en marcha

Para obtener los datos necesarios \hat{e}_{LL} y ω_m se ha optado por realizar 4 mediciones de cada uno de ellos y promediarlas para obtener la constante k_E .

La velocidad será obtenida directamente desde el encoder, midiendo mediante un sensor infrarrojo los intervalos entre ranuras. De esta forma, conociendo el número de ranuras del disco y el periodo de la señal de salida del sensor infrarrojo, se puede obtener la velocidad de salida de forma directa.

En el caso de la tensión entre fases del motor, se opta por medirla mediante un osci-

loscopio digital modelo DSO5102P de Hantek, representado en la imagen 4.5. Este osciloscopio cuenta con la capacidad de registrar los valores medidos mediante USB a través de software en un ordenador. Esta capacidad ha sido utilizada para obtener en un fichero de texto de tipo *.csv los distintos valores de tensión correspondientes a la forma de onda sinusoidal generada por el motor de imanes permanentes.

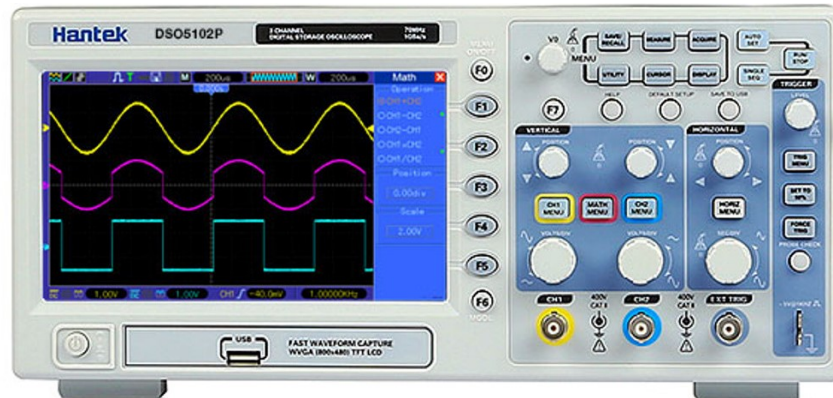


Fig. 4.5: Osciloscopio Hantek DSO-5102P
Fuente: *circuitspecialists.com*

Tras obtener las distintas mediciones, se ha optado por tratar los datos mediante el lenguaje de programación R a través de su entorno de desarrollo RStudio. De esta forma, se ha conseguido automatizar y hacer dinámico el procesado de los datos obtenidos mediante el osciloscopio.

4.1.1.3. Resultados y datos obtenidos

A través de la programación realizada en RStudio, cuyo código puede encontrarse en el Anexo 1, se han analizado los datos tomados desde el osciloscopio logrando así una medida y representación precisas de los valores que nos resultan de utilidad.

El primer valor analizado ha sido el de la tensión de cada una de las 4 ondas sinusoidales muestreadas. Para ello, se han graficado 4 histogramas de cada una de las ondas sinusoidales, lo que permite obtener y validar el valor de tensión de cada onda. Se ha incorporado además en el código un criterio que permite, a discreción del usuario, establecer un criterio de rechazo para las tensiones que se encuentren fuera de un valor predeterminado respecto a la media de la tensión de cada una de las 4 ondas senoidales.

En las gráficas representadas en las imágenes 4.6, 4.7, 4.8 y 4.9, se pueden ver los histogramas correspondientes a las tensiones de pico de cada una de las ondas senoidales analizadas provenientes del motor *brushless*. Sobre cada uno de los histogramas se han representado adicionalmente la distribución normal, distribución de densidad, el valor medio y la mediana; de forma que resulte comparativamente claro analizar los valores de

pico obtenidos para cada una de las ondas.

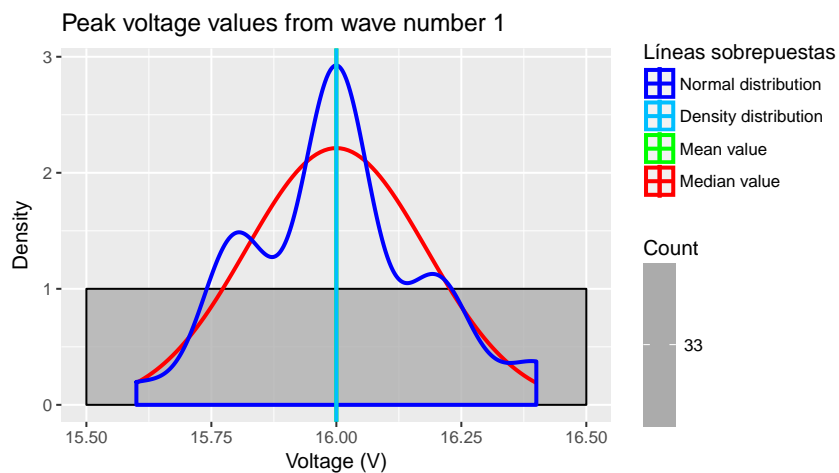


Fig. 4.6: Tensión de pico de la onda senoidal 1

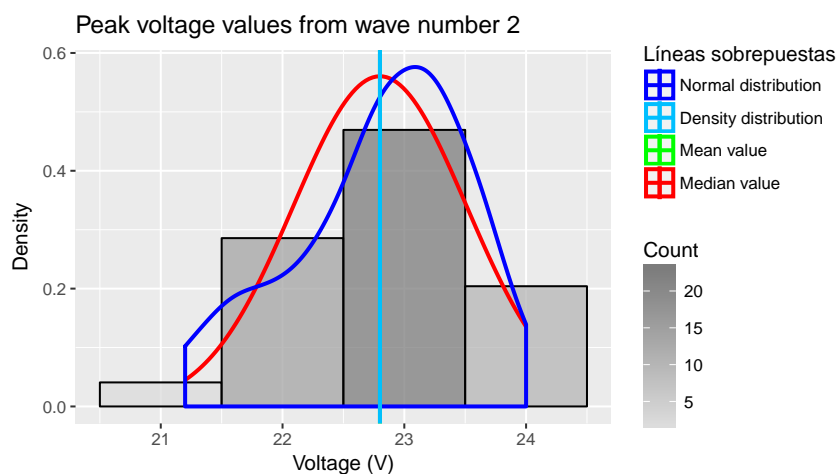


Fig. 4.7: Tensión de pico de la onda senoidal 2

Las formas de onda de todas las señales sinusoidales medidas y almacenadas por el osciloscopio digital, han sido representadas en las figuras presentadas en las imágenes 4.10, 4.11, 4.12 y 4.13, de esta forma se pueden ver las formas de onda y valores de tensión exactamente como fueron medidas en bornes del motor *brushless*.

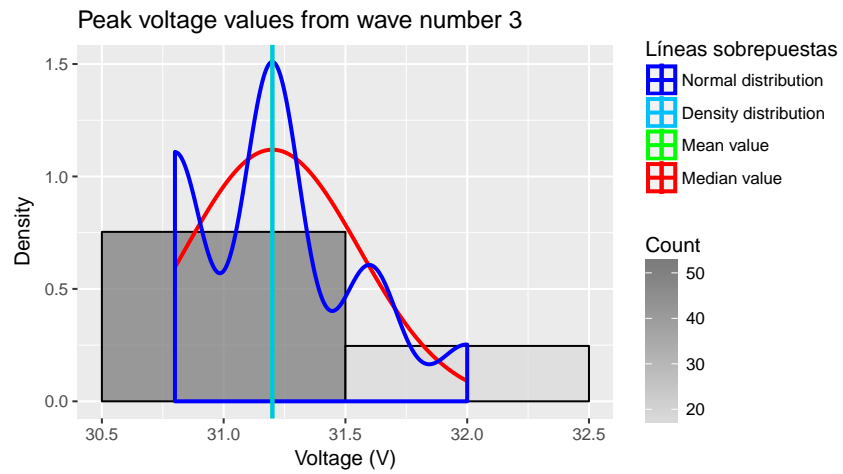


Fig. 4.8: Tensión de pico de la onda senoidal 3

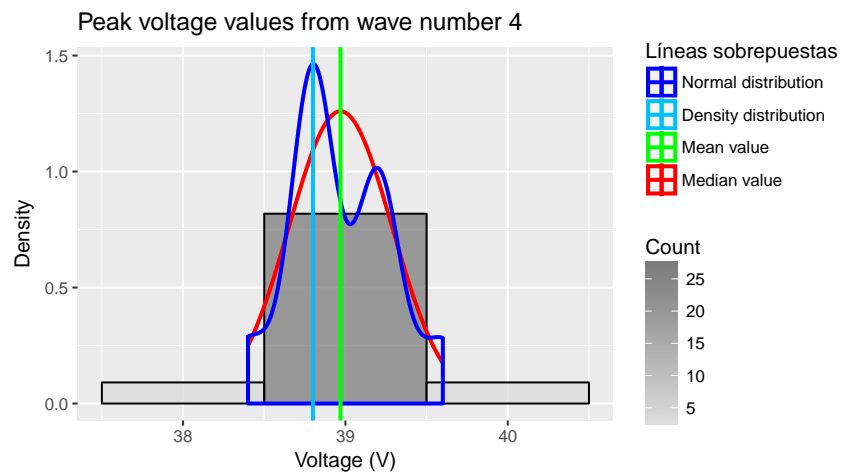


Fig. 4.9: Tensión de pico de la onda senoidal 4

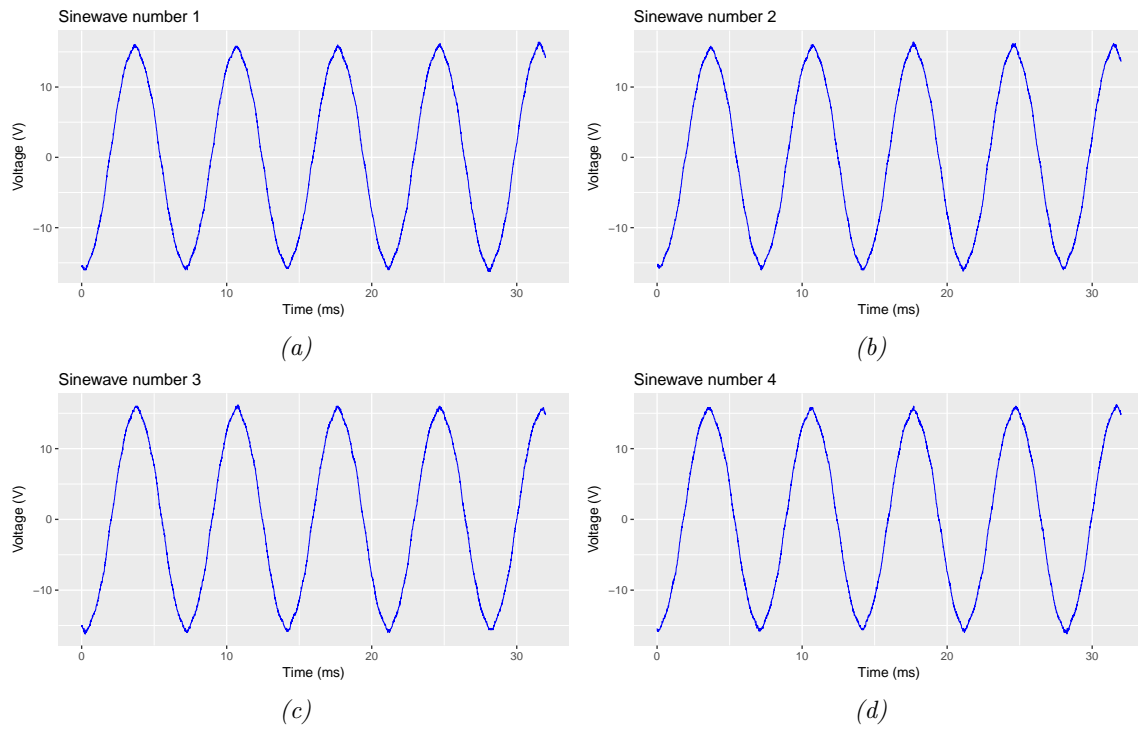


Fig. 4.10: Conjunto de señales senoidales 1

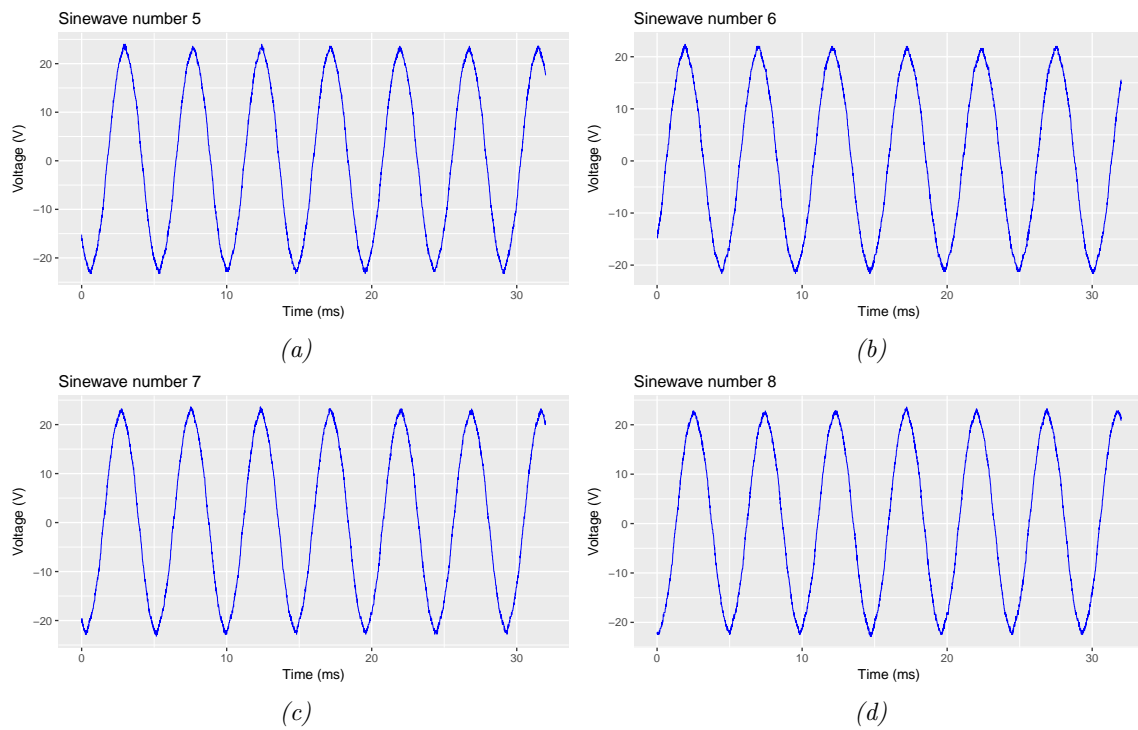


Fig. 4.11: Conjunto de señales senoidales 2

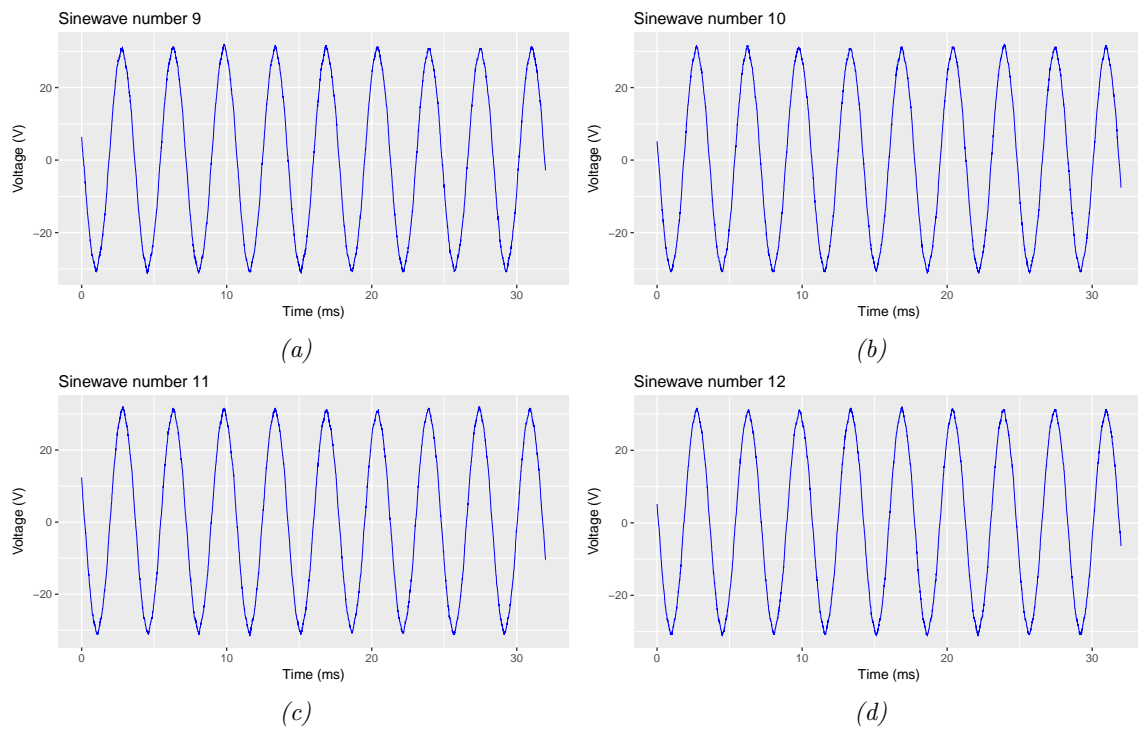


Fig. 4.12: Conjunto de señales senoidales 3

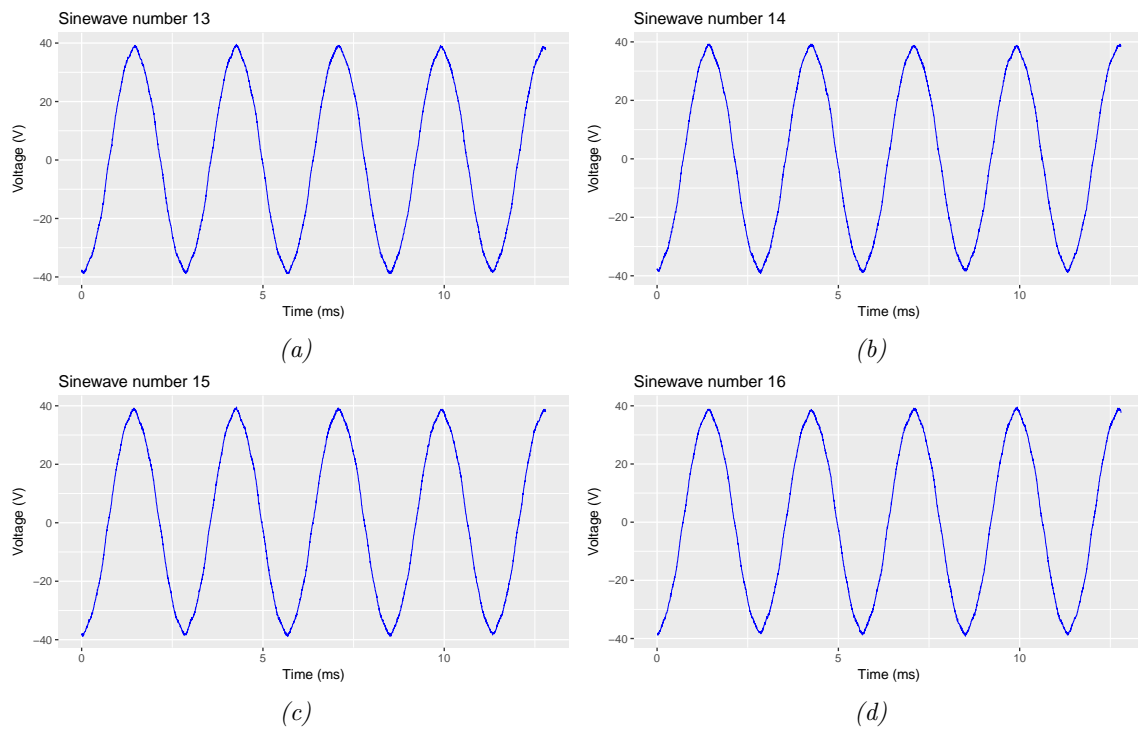
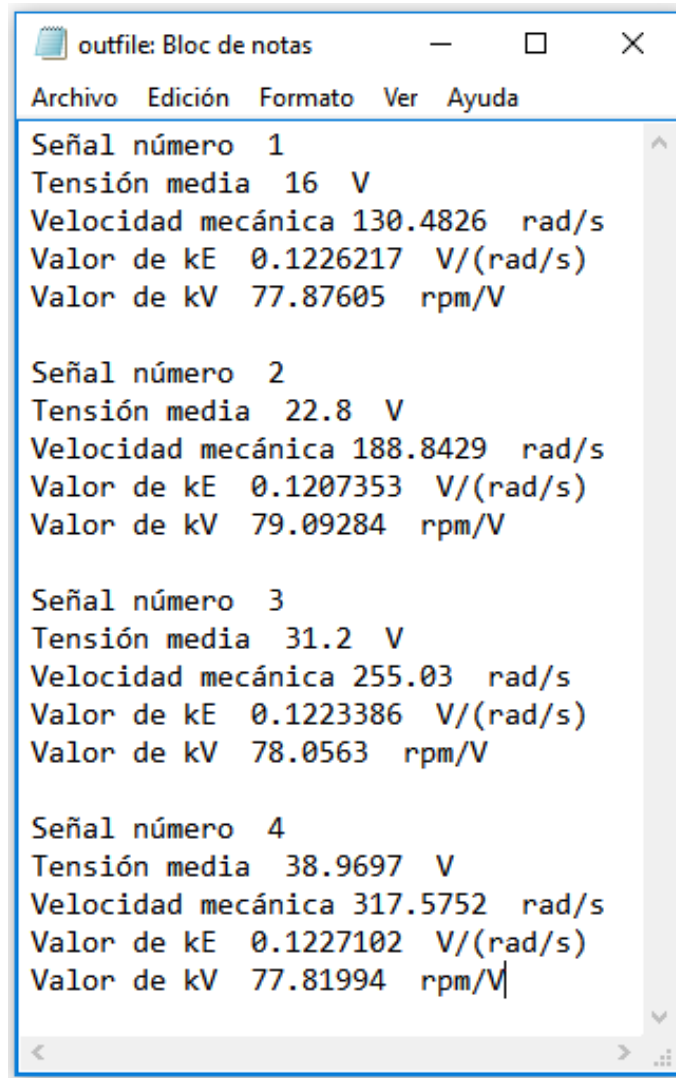


Fig. 4.13: Conjunto de señales senoidales 4

4.1.1.4. Obtención de k_E

Mediante la programación realizada en RStudio y los valores de tensión y velocidad medidos a través del software del osciloscopio, se han podido obtener los datos de tensión y velocidad, así como las constantes k_E y k_V calculadas de forma automática. Los datos de salida han sido almacenados en un fichero de texto de salida desde el propio programa realizado en R, en la figura 4.14 podemos ver una captura de pantalla de los datos obtenidos almacenados en el fichero de texto.



```
outfile: Bloc de notas
Archivo Edición Formato Ver Ayuda
Señal número 1
Tensión media 16 V
Velocidad mecánica 130.4826 rad/s
Valor de kE 0.1226217 V/(rad/s)
Valor de kV 77.87605 rpm/V

Señal número 2
Tensión media 22.8 V
Velocidad mecánica 188.8429 rad/s
Valor de kE 0.1207353 V/(rad/s)
Valor de kV 79.09284 rpm/V

Señal número 3
Tensión media 31.2 V
Velocidad mecánica 255.03 rad/s
Valor de kE 0.1223386 V/(rad/s)
Valor de kV 78.0563 rpm/V

Señal número 4
Tensión media 38.9697 V
Velocidad mecánica 317.5752 rad/s
Valor de kE 0.1227102 V/(rad/s)
Valor de kV 77.81994 rpm/V
```

Fig. 4.14: Captura de pantalla de los datos generados por el programa realizado en R

En la tabla 4.3 encontramos los valores de k_E y k_V calculados mediante el software y las medidas tomadas, junto a los valores nominales aportados por el fabricante.

Como puede observarse, los valores calculados para las constantes se encuentran dentro del porcentaje de error admisible respecto a los valores provistos por el fabricante.

	Valor nominal	Medida 1	Medida 2	Medida 3	Medida 4
k_E ($V/\frac{rad}{s}$)	0.133	0.122	0.121	0.122	0.123
k_V (rpm/V)	72	77.87	79.09	78.05	77.82

Tabla 4.3: Comparación del valor de las constantes

Se puede concluir en este punto, que el ensayo ha sido realizado con éxito, que las mediciones a través del osciloscopio han sido fiables, y que la programación realizada en lenguaje R ha logrado calcular de forma satisfactoria la constante del motor. Rechazando asimismo aquellos valores de tensión anómalos, produciendo un valor para la constante fiable e inmune a medidas anómalas realizadas por el instrumento de medida. Se ha incluido también la capacidad de representar de forma gráfica las formas de onda sinusoidales medidas en bornes del motor, los valores de tensión de pico de dichas ondas, y la generación de un fichero de texto donde se almacene de forma resumida los resultados calculados mediante el programa.

4.2. Encoder óptico

Tras analizar las alternativas planteadas para escoger un elemento de realimentación de la posición, el encoder seleccionado para implementar en la plataforma de ensayo ha sido un encoder óptico. Las ventajas que han logrado que un encoder de tipo óptico fuera el escogido se pueden encontrar a continuación.

- Mayor fiabilidad respecto a un encoder mecánico, nulo mantenimiento al carecer de desgaste interno.
- Robusto frente a entornos industriales adversos, se encuentra encapsulado en un envolvente metálico.
- Su lectura resulta menos intensiva si lo comparamos con un encoder magnético, y su acople más sencillo.

Al comparar un encoder óptico con uno de tipo mecánico resultan claras las ventajas del primero, nulo desgaste y tolerancia a ambientes industriales adversos. Ello consigue que se posicione por encima del encoder de tipo mecánico.

Los encoders magnéticos presentan dificultades para su acople al sistema mecánico, lo que ha llevado a su rechazo para esta aplicación concreta.

Dentro del abanico de posibilidades para escoger un encoder, el modelo seleccionado ha sido el **A38S6**, se trata de un encoder óptico genérico de propósito general cuyo

encapsulado es idéntico al presentado en la imagen 3.17. La señal cuadrada de salida de este encoder presenta una resolución de 600 pulsos por revolución. Al ser leído en cuadratura, tal como se analizó en el planteamiento de alternativas, la máxima resolución alcanzable pasaría a ser de 2400 pulsos por revolución, con capacidad para distinguir el sentido de giro. El encoder escogido presenta asimismo un eje de 6 mm de diámetro al que puede ser acoplado con facilidad cualquier tipo de acople genérico o de diseño propio.

4.3. Microcontrolador STM32F302R8

El microcontrolador escogido ha sido el modelo **STM32F302R8** de STMicroelectronics, cuyas características se encuentran en el apartado alternativas, donde fue presentado como opción alternativa para el control del motor.

Este microcontrolador de 32 bits ha sido escogido por su gran capacidad de cómputo. Si tenemos en cuenta las técnicas de control que pueden llegar a implementarse para el control de un motor *brushless*, resulta evidente que la mayor capacidad de cálculo unida a una mayor velocidad de reloj permite una discretización mayor. Si se optara por un control SPWM este microcontrolador ofrecería una señal más suavizada, debido al mayor número de valores posibles de calcular, lo que provocaría una señal más suave en el motor disminuyendo el rizado de corriente y par.

Existe una placa de desarrollo fabricada por STMicroelectronics que incorpora el microcontrolador escogido, representada en la figura 4.15, la cual representa una opción viable para implementar una solución de desarrollo o pruebas de forma rápida.

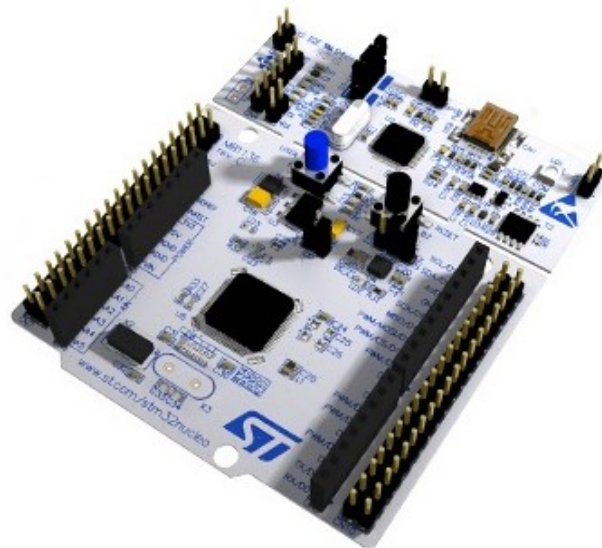


Fig. 4.15: Placa de desarrollo NUCLEO-F302R8

Fuente: st.com

4.4. *Driver* de potencia integrado L6230

El *driver* integrado **L6230** ha sido la opción escogida para la implementación del control de los motores, disponemos de esta forma de todos los elementos que componen el controlador dentro de un mismo circuito integrado con un número limitado de componentes externos. En la tabla 4.4 se han reflejado algunas de las características de ambas implementaciones que han sido consideradas para la elección.

<i>Driver</i> integrado L6230	Implementación con componentes individuales
Tiempos de conmutación mayores	Tiempos de conmutación menores
Inclusión de <i>dead-times</i> integrada en el propio chip	Implementación de <i>dead-times</i> a través de programación en el microcontrolador
Bajo coste de un único elemento	Mayor coste final tras el montaje de todos los elementos

Tabla 4.4: Comparativa entre las posibles implementaciones del *drivers*

Un controlador que se encuentre integrado dentro del mismo chip, presenta ventajas sobre la implementación a partir de componentes individuales que fue descrita al plantear las distintas alternativas. Dentro de ese abanico de ventajas aquellas que han resultado determinantes para la elección de este tipo de controlador han sido las que se indican a continuación.

- La sencillez de control de la lógica de control presente en un *driver* integrado. Dentro de las posibilidades de control, un controlador integrado nos permite el total control de cada una de las partes del semipunto de forma sencilla, partiendo de unas tensiones de control a niveles lógicos provenientes de un microcontrolador.
- Inclusión automática de *dead times* en la secuencia de control. Esto resulta un factor crítico y limitante a la hora de controlar los distintos semipuntos formados por MOSFETs, ya que la inclusión interna de este tipo de tiempos “muertos” permite que las señales del microcontrolador puedan ser lanzadas de forma continua sin presentar una carga de programación mayor al haber de tenerse en cuenta estos tiempos.
- El aumento de fiabilidad al reducir el número de componentes presentes en nuestro circuito. Se trata de un integrado robusto cuyos controles de fabricación son muy estrictos. Se reduce de esta forma la tasa de fallos al no contar con multitud de componentes discretos y semiconductores que podrían desencadenar una tasa de fallos mayor.

El controlador L6230 escogido se puede encontrar ensamblado en una placa de expansión de STMicroelectronics pensada para el uso conjunto con una placa de desarrollo NUCLEO. La placa escogida ha sido el modelo X-NUCLEO-IHM07M1, representada en la figura 4.16, la cual dispone de todos los componentes necesarios ensamblados para poder controlar un motor *brushless* a través del microcontrolador que se encuentre en la tarjeta NUCLEO en la que se instale.



Fig. 4.16: Placa de expansión X-NUCLEO-IHM07M1

Fuente: digikey.co.uk

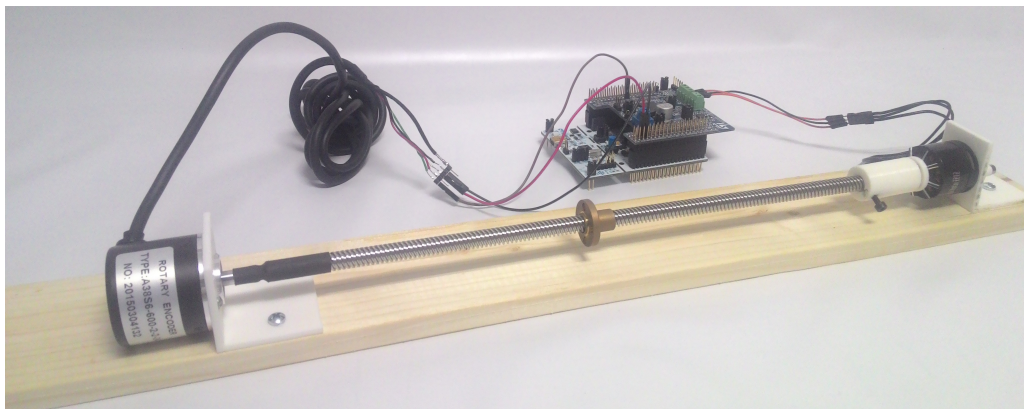
4.5. Plataforma de ensayos

Tras analizar en detalle los elementos que han sido elegidos para formar parte de la solución adoptada, se han implementado en una plataforma de ensayos que permitirá evaluar su funcionamiento.

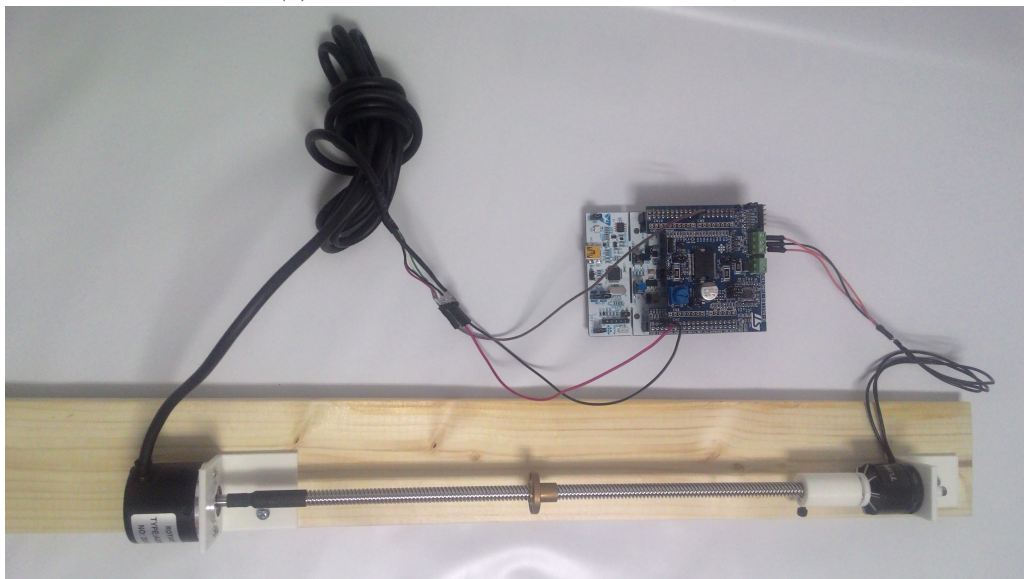
La base de la plataforma está compuesta por un listón de madera de pino, sobre el que se han incorporado dos soportes de plástico impresos en 3D que permiten montar el encoder y el motor. El motor se encuentra acoplado, a su vez, al husillo trapezoidal de 8 mm y 4 entradas mediante una pareja de acoples machihembrados impresos en 3D. En el otro extremo del husillo se encuentra montado el encoder, acoplado al husillo trapezoidal mediante una conexión de tubo termorretráctil, que se comporta como un acople flexible pero que transmite de forma eficiente la rotación bajo condiciones sin carga.

Al tratarse de un sistema libre de cargas y rozamientos, los acoples y la mínima estructura resultan suficientes para soportar los esfuerzos mecánicos sufridos durante el funcionamiento normal de la plataforma. Por ello, se trata además de una plataforma pequeña y ligera que puede ser fácilmente instalada o transportada en cualquier laboratorio.

El montaje final de los elementos mecánicos puede verse en la imagen 4.17(a) y 4.17(b), donde además de encontrarse los elementos electromecánicos acoplados, también se pueden apreciar las conexiones eléctricas necesarias a realizar entre los mismos y la placa de expansión X-NUCLEO-IHM07M1.



(a) Vista lateral de la plataforma de ensayos



(b) Vista superior de la plataforma de ensayos

Fig. 4.17: Diferentes vistas de la plataforma de ensayos

La placa de desarrollo NUCLEO-F302R8 que contiene al microcontrolador, se encuentra situada bajo la placa de expansión que aloja al *driver* de potencia L6230. Es en ésta placa de desarrollo, donde mediante la conexión USB incorporada, debe cargarse la

programación realizada en Atollic TrueSTUDIO.

El programa realizado, debe ser apto para el microcontrolador STM32F302R8. Al tratarse de una plataforma de ensayo, se ha optado por una programación que fuera un compromiso entre potencia de cómputo y rendimiento. El encoder, por tanto, ha sido leído a 600 pulsos por revolución, descargando de esa manera al microcontrolador de la tarea de procesar 2400 pulsos leídos en cuadratura. El control del motor sin escobillas ha sido realizado siguiendo una técnica de control trapezoidal, también denominada *6 step*, consiguiendo reducir así el tiempo de cómputo necesario para excitar cada combinación de semipuentes MOSFET dentro del *driver*.

Capítulo 5

Conclusiones

A continuación se exponen las conclusiones extraídas de la realización del presente proyecto, divididas en conclusiones sobre el propio trabajo, y reflexiones sobre futuras mejoras

5.1. Sobre el trabajo

En este trabajo se han descrito los elementos que componen en la actualidad el sistema de movimiento de una impresora 3D. Se han analizado de forma pormenorizada, valorando cada elemento de forma individual, remarcando sus carencias, para plantear alternativas a los mismos.

La elección de los componentes se ha llevado a cabo mediante una serie de criterios establecidos para lograr resolver los objetivos planteados.

En el caso del motor, se trata de un elemento que ha sido evaluado de forma experimental mediante ensayo para evaluar su elección. El ensayo ha sido desarrollado de forma exhaustiva permitiendo evaluar con precisión los motores ensayados. Lo que ha permitido escoger un motor cuyas características fueran un compromiso entre rendimiento y coste.

Se ha logrado implementar una plataforma de desarrollo con control en bucle cerrado, que une al encoder como elemento de realimentación y al motor de corriente continua sin escobillas como actuador. Permitiendo mediante la programación del microcontrolador de 32 bits, y su actuación sobre el *driver* L6230, el control en bucle cerrado del desplazamiento lineal a través de un husillo.

5.2. Futuras mejoras

Este trabajo significa el planteamiento de la base sobre la que implementar una solución definitiva para una máquina real. Representa por tanto, la primera de las fases que serían

necesarias para implementar el desarrollo expuesto en el presente trabajo en un eje de movimiento lineal real.

Los ensayos realizados, han sido llevados a cabo de forma exhaustiva y prestando especial atención a los datos obtenidos y a su correlación con la realidad. No obstante, estos siempre pueden ser objeto de mejora, logrando una mayor precisión y fiabilidad de los ensayos.

El sistema de control puede ser mejorado tanto como potencia de cómputo ofrezca el microcontrolador. Permitiendo de esta manera llegar a desarrollar algoritmos de control complejos que quedan fuera del ámbito de este proyecto.

Bibliografía

- [1] Aa. Vv., 'About - RepRapWiki', (last modified March, 2017). [Online]. Available: <http://reprap.org/wiki/About>. [Accessed: 03 - Jul - 2017]
- [2] Donald P. Labriola, "Precision motion evolution: Early history of precision motors (1960s to 1980s)", 2002. [Online]. Available: <http://www.machinedesign.com/motorsdrives/precision-motion-evolution-early-history-precision-motors-1960s-1980s>. [Accessed: 11-May-2017].
- [3] J.R. Hendershot and T.J.E. Miller, *Design of brushless permanent-magnet motors*. Oxford, UK: Oxford UP: Magna Physics, 1994.
- [4] Mariss, 'Advantages & Disadvantages of Stepper motors & DC servo motors', NA. [Online]. Available: http://www.machinetoolhelp.com/Automation/systemdesign/stepper_dc servo.html. [Accessed: 05-June-2017]
- [5] Nicholas Morozovsky, Robert Moroto and Thomas Bewley, "RAPID: An Inexpensive Open Source Dynamometer for Robotics Applications", Coordinated Robotics Laboratory, University of California, San Diego, CA, 2013.
- [6] Roger Juanpere Tolrà, "Técnicas de control para motores *Brushless*", Ingenia, Barcelona.
- [7] Tae-Sung Kim, Byoung-Gun Park, Dong-Myung Lee, Ji-Su Ryu, and Dong-Seok Hyun, "A New Approach to Sensorless Control Method for Brushless DC Motors", *International Journal of Control, Automation, and Systems*, vol. 6, no. 4, pp. 477-487, August 2008.
- [8] Takashi Kenjo and S. Nagamori, *Permanent-magnet and brushless DC motors*. Oxford, UK: Clarendon Press: Monographs in electrical and electronic engineering, 1985.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Diseño e implementación de un control en
bucle cerrado de bajo coste para motores de
corriente continua aplicado a impresoras 3D

II - PLIEGO DE CONDICIONES

Valencia, julio de 2017

Autor: D. Eduardo M. Alonso Sánchez

Tutor: D. Ángel F. Perles Ivars

Cotutor: D. Miguel Sánchez López

Pliego de condiciones

1. Definición y alcance del pliego

El presente proyecto supone el diseño e implementación para el testeo de un control en bucle cerrado para un motor de corriente continua sin escobillas, cuya aplicación se centra en cubrir las necesidades de las impresoras 3D autoconstruidas. El trabajo tiene en cuenta las necesidades reales de las máquinas de impresión 3D, a pesar de su carácter académico.

El objeto del presente documento es fijar por tanto los criterios técnicos que permitan la reproducción, desarrollo o implementación del proyecto “Diseño e implementación de un control en bucle cerrado de bajo coste para motores de corriente continua aplicado a impresoras 3D”, de acuerdo a las especificaciones recogidas en este pliego.

El ámbito de aplicación de este documento es extensible a todos los elementos formantes del sistema de desarrollo. Las necesidades particulares podrían suponer un cambio en determinadas condiciones donde se adoptarían soluciones diferentes a las propuestas siempre que su cambio no suponga una disminución del funcionamiento y características del sistema.

2. Condiciones y normas de carácter general

En esta sección se encuentran las condiciones y normas de carácter general.

2.1. Instalación

Se debe comprobar que se cumple con los debidos requisitos para la instalación del *hardware* y *software* necesario.

Para la correcta realización del proyecto se requieren nociones básicas de electrónica y máquinas eléctricas.

2.2. Seguridad

La implementación de cualquier sistema expuesto debe respetar las normas y medidas de seguridad.

Es necesario contar con los debidos elementos de protección, fusibles y diferenciales, en los elementos de alimentación que se estén usando, fuentes de alimentación y/o alimentadores.

La incorrecta implementación puede producir daños personales o materiales. Cualquier responsabilidad que se derive de la alteración, utilización bajo condiciones no controladas, o uso irresponsable, recae exclusivamente en el usuario.

2.3. Utilización

La utilización de la plataforma de desarrollo queda sujeta a las indicaciones de los fabricantes en las hojas de características presentes en los **Anexos**.

2.4. Mantenimiento

De forma previa a cada utilización y como norma general tras cada uso de la plataforma de pruebas, ésta debe ser inspeccionada de forma ocular en busca de defectos de tipo mecánico. Se debe limpiar cuando lo requiera para garantizar un buen funcionamiento.

Tras cada prueba y cuando vaya a darse un periodo de inactividad debe desconectarse la alimentación que se esté usando.

Revisar antes de cada puesta en marcha la plataforma de pruebas en busca de fallos de conexionado y obstrucciones mecánicas.

3. Condiciones particulares

3.1. Técnicas

3.1.1. Microcontrolador

Las características técnicas del microcontrolador STM32F302R8 pueden encontrarse en el **Anexo 2**.

Las características técnicas de la placa de desarrollo NUCLEO-F302R8, donde se encuentra implementado el microcontrolador STM32F302R8, se pueden encontrar en el **Anexo 3**.

Este producto cumple con las siguientes normas de seguridad para equipos eléctricos de medida:

- IEC 61010-1, EN 61010-1
- UL 61010-1, CSA 61010-1
- UNE-EN 60745
- UNE-EN 61800

La placa de desarrollo NUCLEO-F302R8 no se encuentra certificada para su uso en ambientes peligrosos.

Control de calidad

Se debe comprobar la serigrafía del integrado montado en la placa de desarrollo, para comprobar de forma ocular que se cuenta con la versión correcta del microcontrolador aquí descrito.

3.1.2. *Driver* de potencia

Las características técnicas del *driver* L6230 se encuentran recogidas en el **Anexo 4**.

Las características técnicas de la placa de expansión X-NUCLEO-IHM07M1, en la que se encuentra implementado el integrado L6230, pueden encontrarse en el **Anexo 5**.

Control de calidad

Debe comprobarse la serigrafía del integrado incorporado en la placa para asegurarse de forma ocular de que se cuentan con el integrado indicado.

Se comprobarán los niveles de tensión de salida de los reguladores así como el buen funcionamiento de las clemas atornillables incorporadas.

3.1.3. Motor

Las características del motor TURNIGY HD-2212 se encuentran relacionadas en la tabla 1.

Control de calidad

Debe comprobarse el buen estado físico del motor, y mediante un giro manual evaluar la suavidad con que se produce el giro sin roces mecánicos.

Comprobar el movimiento del eje alimentado con un inversor para comprobar el correcto funcionamiento del motor en giro libre, sin carga.

Comprobar la forma de onda de la fuerza electromotriz con un osciloscopio para su verificación.

Fabricante	TURNIGY
Modelo	HD-2212
Diámetro del motor	27 mm
Tensión nominal	12 V
Número de bobinados	12 (4 arrollamientos por fase)
Número de polos	14 (7 pares de polos)
Número de espiras por bobina	12 V
k_E	$0.133 \text{ V}/\frac{\text{rad}}{\text{s}}$ ($k_V = 72 \text{ rpm}/\text{V}$)
k_T	$0.133 \text{ N}\cdot\text{m}/\text{A}$

Tabla 1: Características del motor TURNIGY HD-2212

3.1.4. Encoder

Las características del encoder A38S6 pueden encontrarse relacionadas en la tabla 2.

Modelo	A38S6
Tensión de alimentación	5-30 V
Tipo de blindaje	Apantallado
Número de salidas	2 (“A” y “B”)
Tipo de salida	Cuadratura

Tabla 2: Características del encoder A38S6

Control de calidad

Debe comprobarse el buen estado físico del encoder, y mediante un giro manual comprobar la suavidad con que se lleva a cabo el giro del mismo, los rodamientos garantizan un rozamiento nulo que debe ser comprobado.

Comprobar la salida del encoder con un osciloscopio bajo tensiones de alimentación nominales y comprobar su buen funcionamiento.

3.1.5. Piezas impresas en 3D

Las piezas necesarias para el montaje han sido realizadas mediante una impresora 3D de tipo Mendel PRUSA i3, cuyas características de impresión han sido: altura de capa de 0.2 mm, relleno del 25 % y 4 capas superiores/inferiores sólidas.

Para la unión de los distintos elementos que precisan el uso de tornillería adicional, se ha utilizado tornillería métrica M3 y rosca-chapa de 2.5 mm. Toda la tornillería utilizada es de acero cincado resistente a la corrosión.

Control de calidad

Debe comprobarse mediante el uso de un calibre las dimensiones y tolerancias de las piezas impresas, así como realizar una evaluación ocular para evaluar el correcto materializado de las mismas.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Diseño e implementación de un control en
bucle cerrado de bajo coste para motores de
corriente continua aplicado a impresoras 3D

III - PRESUPUESTO

Valencia, julio de 2017

Autor: D. Eduardo M. Alonso Sánchez

Tutor: D. Ángel F. Perles Ivars

Cotutor: D. Miguel Sánchez López

Presupuesto

1. Detalles del presupuesto

Para la elaboración del presupuesto se han tomado los precios de los catálogos proporcionados por los fabricantes así como los precios ofertados en las tiendas *online* que actúan como distribuidoras de los mismos, obteniendo de esta manera el precio final de adquisición para el consumidor final de todos los productos.

El coste de la mano de obra se ha calculado a basándose en la estimación del salario horario de un graduado en ingeniería técnica industrial (27,5 €). Para ello se ha tenido en cuenta la dedicación estimada obtenida conforme a la carga de 12 ECTS correspondientes al presente proyecto, donde 1 ECTS corresponde a 25 horas. La dedicación total asciende a 300 horas.

Se ha añadido un porcentaje de gastos del 10 por ciento, contemplando de esta manera los gastos generales en que se pueda incurrir durante la realización de la plataforma de desarrollo.

2. Materiales y componentes

Materiales y componentes				
Uds.	Denominación	Cantidad	Precio (€)	Total (€)
u	Encoder A38S6	1	12,05	12,05
u	Husillo 8mm / 4 entradas	1	3,88	3,88
u	Motor HD-2212	1	16,03	16,03
u	NUCLEO-F302R8	1	9,61	9,61
u	X-NUCLEO-IHM07M1	1	11,67	11,67
u	Soportes impresos en 3D	2	5,00	10,00
u	Acoples impresos en 3D	2	3,00	6,00
u	Tornillería variada	10	0,1	1,00
Subtotal materiales y componentes				70,24

Tabla 1: Coste de materiales y componentes

3. Mano de obra

Mano de obra				
Uds.	Denominación	Cantidad	Precio (€)	Total (€)
h	Investigación, diseño y desarrollo	300	27,5	8250
Graduado en Ingeniería Electrónica Industrial y Automática				
Subtotal mano de obra				8.250,00

Tabla 2: Coste de mano de obra

4. Coste del proyecto

Coste del proyecto	
Materiales y componentes	70,24
Mano de obra	8.250,00
TOTAL	8.320,24
IVA 21 %	1747,25
Gastos generales 10 %	832,02
PVP	10.899,51

Tabla 3: Coste de mano de obra

El precio total es de 10.899,51 €.- diez mil ochocientos noventa y nueve euros con cincuenta y un céntimos.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Diseño e implementación de un control en
bucle cerrado de bajo coste para motores de
corriente continua aplicado a impresoras 3D

IV - ANEXOS

Valencia, julio de 2017

Autor: D. Eduardo M. Alonso Sánchez

Tutor: D. Ángel F. Perles Ivars

Cotutor: D. Miguel Sánchez López

Anexo 1

Código en lenguaje R para el tratamiento de señales

A continuación podemos encontrar el código en lenguaje R, usado para tratar y graficar los datos correspondientes a las ondas senoidales del motor *brushless*, mediante los datos en formato *.csv obtenidos a través del osciloscopio DSO5102P.

```
1 # -----
2 # LOAD NEEDED LIBRARIES
3 # -----
4
5 library(ggplot2)
6
7 # -----
8 # VARIABLES THAT CAN BE MANIPULATED
9 # -----
10
11 # Number of .csv per wave, specify each wave sampled .csv number
12 #   in order
13 # *.csv's from the same wave should be in the same folder
14 # [ e.g. 1st wave 4 .csv files, 2nd wave 5 .csv files, 3rd wave 3
15 #   .csv files
16 #   for the provided example nfiles should be c(4,5,3) ]
17 nfiles <- c(4,4,4,4)
18
19 # Percentage of the max stored peak to consider we have a 'zero
20 #   cross' detection
21 # must be written as: threshold = percentage/100 (e.g. 20% =>
22 #   threshold <- 0.2)
23 threshold <- 0.1
```

```
20
21 # Number of samples averaged to evaluate the 'zero cross'
    detection
22 l <- 8
23
24 # Number of pulses per encoder (complete periods per revolution)
25 encoder_pulses <- 4
26
27 #####
28 # NO VARIABLES CAN BE MODIFIED BELOW THIS POINT
29 #           CODE STARTS HERE
30 #####
31
32 # -----
33 # LOAD THE *.csv FILES AND STORE THEIR VALUES
34 # -----
35
36 # List files in the encoder directory including path in the name
    using full.names = TRUE
37 file_names_enc <- list.files(path = "./enc/", full.names = TRUE)
38
39 # List files in the oscilloscope directory including path in the
    name using full.names = TRUE
40 file_names_osc <- list.files(path = "./osc/", full.names = TRUE,
    recursive = TRUE)
41
42 # Create the dataframes
43 data_enc <- list()
44 data_osc <- list()
45
46 # Read .csv files and add data to the dataframes
47 data_enc <- lapply(file_names_enc, read.csv)
48 data_osc <- lapply(file_names_osc, read.csv)
49
50 # Create the column names
51 colnames_enc <- c("tiempo", "canal_0")
52 colnames_osc <- c("tiempo", "tension")
53
54 # Add column names to dataframes
55 data_enc <- lapply(data_enc, setNames, colnames_enc)
```

```
56 data_osc <- lapply(data_osc, setNames, colnames_osc)
57
58 # Close the file where the output log is written, to erase old
    data stored
59 close( file( "figs/outfile.txt", open="w" ) )
60
61 # Replace every 'time' column values with the fitting sequence
    starting
62 # with 0 and with each step being equal to time between samples
63 for( h in 1:length(data_osc) ){
64
65     # We take the time between samples subtracting 19th sample to
        20th sample
66     # (using samples number 19 and 20 at random)
67     pass <- data_osc[[h]][[1]][[20]] - data_osc[[h]][[1]][[19]]
68
69     # Generate the total sampling time = time between samples *
        number of samples
70     ttime <- pass * length(data_osc[[h]][[1]])
71
72     # Replace time column with the fitting sequence
73     data_osc[[h]][[1]] <- seq(0,ttime - pass,by = pass)
74     # data_osc <- lapply(data_osc, function(data_osc){data_osc[,1]
        <- g; data_osc})
75
76     # Increase h
77     h <- h+1
78
79 }
80
81 # Get number of sampled waves (data frames)
82 nwaves <- length(nfiles)
83
84 # Value being read in the actual loop iteration
85 actual_value <- 0
86
87 # Stored peak, maximum peak found until it's definitely stored,
88 # (when a lesser value is found)
89 stor_peak <- 0
90
```



```
91 # Zero cross detection average, averaged value to evaluate if we
    # crossed the zero
92 zero_avg <- 0
93
94 # First peak
95 fpeak <- 0
96
97 # Incremental for definitive stored maximums
98 # (just being used to make max_peaks grow)
99 peakscout <- 0
100
101 # Create list to store output maximums found dynamically
102 max_peaks <- lapply( 1:nwaves, function(z) z )
103
104 # Create list to store semiperiods time dynamically
105 semiperiodos <- lapply( 1:nwaves, function(z) z )
106
107 # List to store speeds in rev/s
108 w <- list()
109
110 # List to store kE constant in V/(rad/s)
111 kE <- list()
112
113 # List to store kV constant in rpm/V
114 kV <- list()
115
116 actual_time <- 0
117 periodcount <- 0
118
119 rdata <- 0
120
121 k <- 0
122
123 # Big main loop to move around all sampled files from all waves
124 for( f in 1:nwaves ){
125
126     # First loop, one for each *.csv file
127     for( j in 1:nfiles[f] ){
128
129         # First we need to determine which data frame we are going to
```

```
    read
130  rdata <- sum(nfiles[1:(f-1)])+j
131
132  if(f == 1){
133    rdata <- j
134  }
135  # Print each read dataframe, just to check the number of
    times it has been done
136  cat("rdata:␣", rdata, "\n")
137
138  # Get length of actual loop from 2nd column
139  n <- length(data_osc[[rdata]][[2]])
140
141  # Second loop, one for each element 'n' (~3200)
142  for( m in 1:n ){
143
144    # Add actual value to zero_avg
145    zero_avg <- zero_avg + abs(data_osc[[rdata]][[2]][[m]])
146
147    # Store the actual value being read
148    actual_value <- abs(data_osc[[rdata]][[2]][[m]])
149
150    # Store in stor_peak the maximum value found until a zero
    cross is detected
151    if( actual_value > stor_peak ){
152
153      stor_peak <- actual_value
154
155    }
156
157    # Check 'l' averaged values for zero cross detection
158    if( (k == 1) ){
159
160      # Compare averaged value of 'l' elements with threshold
    percentage*temporal maximum
161      if( (zero_avg/l) < (stor_peak*threshold) ){
162
163        # Store the temporal maximum definitely
164        max_peaks[[f]][[length(max_peaks[[f])+peakcount]] <-
          stor_peak
```

```
165
166     if( peakscout == 0){
167         peakscout <- 1
168     }
169
170     stor_peak <- 0
171
172 }
173
174 zero_avg <- 0
175
176 k <- 0
177
178 periodcount <- 1
179
180 }
181
182 k <- k+1
183 m <- m+1
184
185 }
186
187 peakscout <- 0
188 j <- j+1
189
190 # Reset first peak found variable to default value '0'
191 fpeak <- 0
192
193 }
194
195 # We could expect this line down there to be uncommented, but
196 # doing so we keep
197 # 'j' value from previous loop so it grows
198 j <- 0
199
200 # -----
201 # ENCODER READING STARTS HERE
202 # -----
203
204 # Set to zero both times and extra count to be used later
```

```
204 tiempoa <- 0
205 tiempob <- 0
206
207 tiempos_count <- 0
208
209 # Create a temporal storage for the period times
210 tmp_time <- matrix(0, ncol = 1, nrow = ( length(data_enc[[f
      ]][[1]])/2)-1 )
211
212 # Second loop, one for each value to be STORED, so (length/2)
      -1, see above
213 for( y in 1:length(data_enc[[f]][[1]]) ){
214
215     # Store the period time when encoder value is '0'
216     # This way we have the complete period time
217     if( data_enc[[f]][[2]][[y]] == 0 ){
218
219         # Get actual time
220         tiempoa <- data_enc[[f]][[1]][[y]]
221
222         # Substract previous time to get the period (actual -
          previous)
223         tmp_time[(tiempos_count),1] <- tiempoa - tiempob
224
225         # Store the actual time to be used as previous time in the
          next iteration
226         tiempob <- tiempoa
227
228         tiempos_count <- tiempos_count + 1
229
230     }
231
232 }
233
234 # Save as *.pdf
235 plot_name = paste("figs/periodshistogram", f, ".pdf", sep = "")
236
237 # Doing that this way we need dev.off() when plotting is
      finished
238 pdf(plot_name, width=16/2.54, height=9/2.54)
```

```
239
240 # Histogram for encoder mean period times, with prob=TRUE to
      show probabilities not counts
241 hist(tmp_time[-c(1)], prob = TRUE, col = "grey",
242       main = paste("Speed_periods_(sample_number)", f, ")"), sep
      = ""),
243       xlab = "Time_(s)", ylab = "Density" )
244
245 # Add a density estimate line
246 lines(density(tmp_time[-c(1)]), col = "blue", lwd = 2)
247
248 # Density smoothed line, dotted style
249 lines(density(tmp_time[-c(1)]), adjust = 2), lty = "dotted", col
      = "red3", lwd = 2)
250
251 # Mean value vertical line, dashed style
252 abline(v = mean(tmp_time[-c(1)]), col = "green4", lty = "dashed
      ", lwd = 2)
253
254 abline(v = median(tmp_time[-c(1)]), col = "blue", lwd = 2)
255
256 # Finish the *.pdf file saving
257 dev.off()
258
259 # Store the period times definitely in another variable
260 # [tmp_time will be used again in the next iteration temporary]
261 assign(paste("tiempo", f, sep=""), tmp_time)
262
263 tiempos_count <- 0
264
265 # -----
266 # ENCODER READING ENDS HERE
267 # -----
268
269 # Calculate omega (mechanical speed), in rev/s
270 w[[f]] <- ( (1/encoder_pulses)/mean(tmp_time[-c(1)]) )
271 print(paste("Velocidad:", w[f], " rev/s", sep = ""))
272
273 }
274
```

```
275 # -----
276 # PLOT EACH READ SINEWAVE AND PRINT INTO A FILE
277 # -----
278
279 for( popop in 1:length(data_osc) ){
280
281   # Plot title to be added later
282   plot_title = paste("Sinewave□number□", popop, sep = "")
283
284   #plot_y1 = paste("Tensin (V) - ", sep = "")
285   #plot_y2 = paste(" = ", format(round(mean(max_peaks[[j]][2:(
286     length(max_peaks[[j]])-1)]), 2), nsmall = 2), sep = "")
287   #meddd <- format(round(mean(max_peaks[[j]][2:(length(max_peaks
288     [[j]])-1)]), 2), nsmall = 2)
289
290   # Plot the given data by data_osc, aka sinewaves
291   ggplot( data_osc[[popop]], aes(x=tiempo,y=tension))+
292     geom_line(size = 0.25, color = "blue") +
293     xlab("Time□(ms)") +
294     ylab("Voltage□(V)") +
295     ggtitle(plot_title)
296
297   # Save as *.pdf
298   plot_name = paste("figs/sine", popop, ".pdf", sep = "")
299   ggsave( file = plot_name, dpi=600, width = 16, height=9, units=
300     c("cm") )
301
302   # Save as *.jpeg
303   # plot_name = paste("figs/jpeg", popop, ".jpg", sep = "")
304   # ggsave( file = plot_name, dpi=600, width = 16, height=9,
305     units=c("cm") )
306
307 }
308
309 # -----
310 # FILTER ALL THE PEAKS STORED, REJECTING OR STORING THEM
311 #     DEPENDING ON THEIR DEVIATION FROM THE MEAN
312 # -----
313
314 # Create the list to store the definitive voltage values,
```

```
    filtered from max_peaks
311 max_peaks_def <- lapply( 1:nwaves, function(z) z )
312
313 # Variable used to fill the list increasing its index by one
314 maxcount <- 0
315
316 # First loop, one for each group of peaks stored
317 for( p in 1:length(max_peaks) ){
318
319     # Mean of the read max_peaks group
320     mean_max <- mean(max_peaks[[p]])
321
322     for( q in 1:length(max_peaks[[p]]) ){
323
324         # We consider a 10% threshold for the peak being compared
325         # with the mean
326         # Values [over 90% / below 110%] of the mean are OK
327         if( (max_peaks[[p]][[q]] > (mean_max*0.9)) && (max_peaks[[p]
328             ][[q]] < (mean_max*1.1)) ){
329
330             max_peaks_def[[p]][[length(max_peaks_def[[p]])+maxcount]]
331             <- max_peaks[[p]][[q]]
332             #print(max_peaks_def[[p]][[length(max_peaks_def[[p]])]])
333
334             # Increase the length of the index for the variable to be
335             # stored
336             if( maxcount == 0){
337                 maxcount <- 1
338             }
339
340         }
341
342     }
343
344 # Reset this variable each time a complete set of wave values
345 # are processed
346 maxcount <- 0
347
348 # Calculate KE with the filtered peak values in V/(rad/s) and
349 # print
```

```

344 kE[[p]] <- ( mean(max_peaks_def[[p]])/(2*pi*w[[p]]) )
345 print(paste("V/w= $\omega$ kE $\omega$ -> $\omega$ ", mean(max_peaks_def[[p]]), "/", w[[p]
    ], " $\omega$ = $\omega$ ", kE[[p]], "V/(rad/s)", sep = ""))
346
347 # Calculate kV in rpm/V for comparison purposes and print
348
349 kV[[p]] <- ( (60*w[[p]])/mean(max_peaks_def[[p]]) )
350 print(paste("kV $\omega$ = $\omega$ ", kV[p], " $\omega$ rpm/V", sep = ""))
351
352 }
353
354 # -----
355 # PLOT AN HISTOGRAM FOR EACH SINEWAVE'S PEAKS
356 # (After filtering peak values)
357 # -----
358
359 for( ruben in 1:length(max_peaks_def) ){
360
361   plot_name <- paste("Peak $\omega$ voltage $\omega$ values $\omega$ from $\omega$ wave $\omega$ number $\omega$ ",
     ruben, sep = "")
362
363   df <- as.data.frame(max_peaks_def[[ruben]])
364
365   gg <- ggplot(df, aes(x=df))
366
367   gg <- gg + geom_histogram(binwidth=1, colour="black", aes(y=..
     density.., fill=..count..), alpha = 0.75) + guides(order = 1)
368
369   gg <- gg + labs(title=plot_name) + labs(x="Voltage $\omega$ (V)", y="
     Density")
370
371   gg <- gg + scale_fill_gradient("Count", low="#DCDCDC", high="#7
     C7C7C")
372   # Normal distribution - RED
373   gg <- gg + stat_function(fun=dnorm, aes(color = 'red'), args=
     list(mean=mean(df[[1]]), sd=sd(df[[1]])), lwd = 1)
374   # Density - BLUE
375   gg <- gg + geom_density(aes(color = 'blue'), lwd = 1)
376   # Mean - GREEN
377   gg <- gg + geom_vline(aes(xintercept = mean(df[[1]]), color = '

```



```
    green'), lwd = 1)
378 # Median - SKY BLUE
379 gg <- gg + geom_vline(aes(xintercept = median(df[[1]]), color =
    'deepskyblue'), lwd = 1)
380
381 gg <- gg + scale_colour_manual("Lneas sobrepuestas", values = c
    ('red'='red', 'blue'='blue', 'green'='green', 'deepskyblue'='
    deepskyblue'), labels = c('Normal distribution', 'Density
    distribution', 'Mean value', 'Median value'))
382
383 print(gg)
384 #print(ruben)
385
386 # Save as *.pdf
387 plot_name = paste("figs/peakshistogram", ruben, ".pdf", sep = "
    ")
388 ggsave( file = plot_name, dpi=600, width = 16, height=9, units=
    c("cm") )
389
390 }
391
392 # Close all the *.pdf's storage devices
393 dev.off()
394
395 for( storage in 1:length(data_enc) ){
396
397     sink("figs/outfile.txt", append = TRUE)
398     cat("Seal_nmero", storage)
399     cat("\n")
400     cat("Tensin_media", mean(max_peaks_def[[storage]]), "V")
401     cat("\n")
402     cat("Velocidad_mecnica", (w[[storage]]*2*pi), "rad/s")
403     cat("\n")
404     cat("Valor_de_kE", kE[[storage]], "V/(rad/s)")
405     cat("\n")
406     cat("Valor_de_kV", kV[[storage]], "rpm/V")
407     cat("\n\n")
408     sink()
409
410 }
```

411 `sink()`

Anexo 2

En las páginas siguientes puede encontrarse un extracto de las características definitorias del microcontrolador STM32F302R8 obtenidas desde la hoja de características (*datasheet*), descargada de la página web del fabricante STMicroelectronics.

Han sido incluidas las páginas en las que figuran las características definitorias del dispositivo, no obstante, la documentación y hoja de características completa cuenta con un número de páginas extenso donde se detalla de forma pormenorizada cada una de las características de este elemento.

Si se desea consultar la documentación de forma completa, se recomienda acceder a través del portal web del fabricante *www.st.com*.



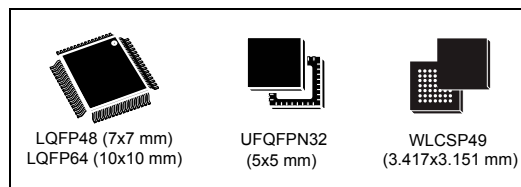
STM32F302x6 STM32F302x8

ARM® Cortex®-M4 32-bit MCU+FPU, up to 64 KB Flash, 16 KB SRAM, ADC, DAC, USB, CAN, COMP, Op-Amp, 2.0 - 3.6 V

Datasheet - production data

Features

- Core: ARM® 32-bit Cortex®-M4 CPU with FPU (72 MHz max.), single-cycle multiplication and HW division, DSP instruction
- Memories
 - 32 to 64 Kbytes of Flash memory
 - 16 Kbytes of SRAM on data bus
- CRC calculation unit
- Reset and power management
 - V_{DD}, V_{DDA} voltage range: 2.0 to 3.6 V
 - Power-on/Power down reset (POR/PDR)
 - Programmable voltage detector (PVD)
 - Low-power: Sleep, Stop, and Standby
 - V_{BAT} supply for RTC and backup registers
- Clock management
 - 4 to 32 MHz crystal oscillator
 - 32 kHz oscillator for RTC with calibration
 - Internal 8 MHz RC with x 16 PLL option
 - Internal 40 kHz oscillator
- Up to 51 fast I/O ports, all mappable on external interrupt vectors, several 5 V-tolerant
- Interconnect matrix
- 7-channel DMA controller supporting timers, ADCs, SPIs, I²Cs, USARTs and DAC
- 1 × ADC 0.20 μs (up to 15 channels) with selectable resolution of 12/10/8/6 bits, 0 to 3.6 V conversion range, single ended/differential mode, separate analog supply from 2.0 to 3.6 V
- Temperature sensor
- 1 × 12-bit DAC channel with analog supply from 2.4 to 3.6 V
- Three fast rail-to-rail analog comparators with analog supply from 2.0 to 3.6 V
- 1 × operational amplifier that can be used in PGA mode, all terminal accessible with analog supply from 2.4 to 3.6 V



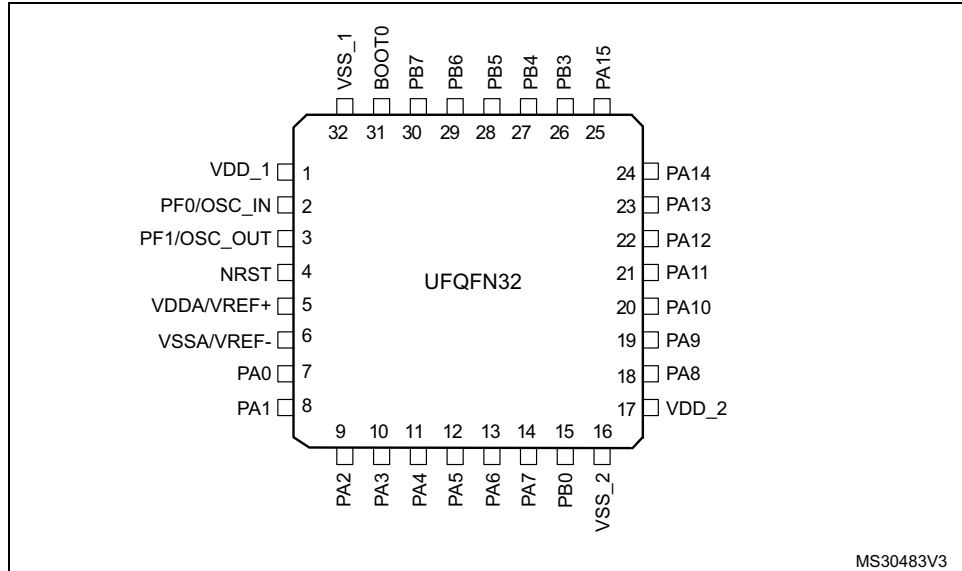
- Up to 18 capacitive sensing channels supporting touchkey, linear and rotary sensors
- Up to 9 timers
 - One 32-bit timer with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
 - One 16-bit 6-channel advanced-control timer, with up to 6 PWM channels, deadtime generation and emergency stop
 - Three 16-bit timers with IC/OC/OCN or PWM, deadtime gen. and emergency stop
 - One 16-bit basic timer to drive the DAC
 - 2 watchdog timers (independent, window)
 - SysTick timer: 24-bit downcounter
- Calendar RTC with alarm, periodic wakeup from Stop/Standby
- Communication interfaces
 - Three I²Cs with 20 mA current sink to support Fast mode plus
 - Up to 3 USARTs, 1 with ISO 7816 I/F, auto baudrate detect and Dual clock domain
 - Up to two SPIs with multiplexed full duplex I²S
 - USB 2.0 full-speed interface
 - 1 × CAN interface (2.0B Active)
 - Infrared transmitter
- Serial wire debug (SWD), JTAG
- 96-bit unique ID

Table 1. Device summary

Reference	Part number
STM32F302x6	STM32F302R6, STM32F302C6, STM32F302K6
STM32F302x8	STM32F302R8, STM32F302C8, STM32F302K8

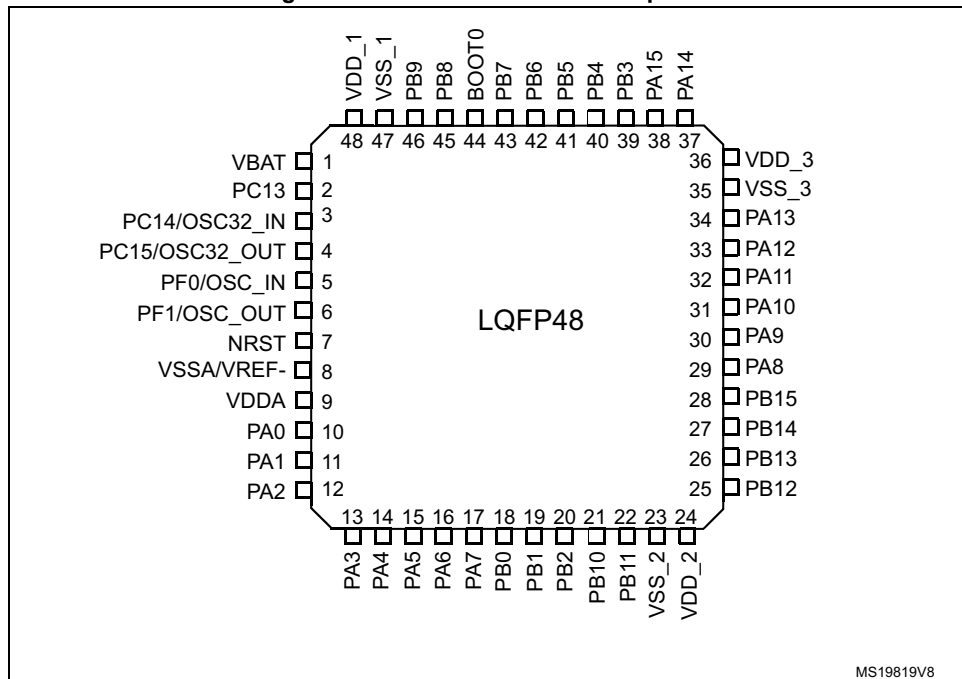
4 Pinouts and pin description

Figure 4. STM32F302x6/8 UFQFN32 pinout



1. The above figure shows the package top view.

Figure 5. STM32F302x6/8 LQFP48 pinout



1. The above figure shows the package top view.

Anexo 3

En las páginas siguientes puede encontrarse un extracto de las características definitivas de la placa de desarrollo NUCLEO-F302R8 obtenidas desde la hoja de características (*datasheet*), descargada de la página web del fabricante STMicroelectronics.

Han sido incluidas las páginas en las que figuran las características definatorias del dispositivo, no obstante, la documentación y hoja de características completa cuenta con un número de páginas extenso donde se detalla de forma pormenorizada cada una de las características de este elemento.

Si se desea consultar la documentación de forma completa, se recomienda acceder a través del portal web del fabricante *www.st.com*.



NUCLEO-XXXXRX

STM32 Nucleo-64 board

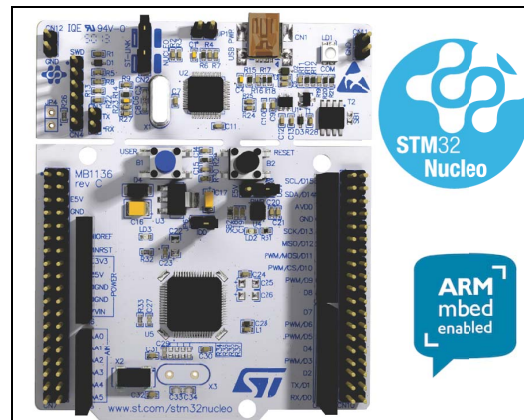
Data brief

Features

- STM32 microcontroller in QFP64 package
- Two types of extension resources:
 - Arduino™ Uno V3 connectivity
 - ST morpho extension pin headers for full access to all STM32 I/Os
- ARM® mbed™ (see <http://mbed.org>)
- On-board ST-LINK/V2-1 debugger/programmer with SWD connector:
 - Selection-mode switch to use the kit as a standalone ST-LINK/V2-1
- Flexible board power supply:
 - USB VBUS or external source (3.3V, 5V, 7 - 12V)
 - Power management access point
- Three LEDs:
 - USB communication (LD1), user LED (LD2), power LED (LD3)
- Two push-buttons: USER and RESET
- USB re-enumeration capability. Three different interfaces supported on USB:
 - Virtual COM port
 - Mass storage
 - Debug port
- Support of wide choice of Integrated Development Environments (IDEs) including IAR™, ARM® Keil®, GCC-based IDEs

Description

The STM32 Nucleo board provides an affordable and flexible way for users to try out new concepts and build prototypes with the STM32 microcontroller, choosing from the various combinations of performance, power consumption and features. The Arduino™ Uno V3 connectivity support and the ST morpho headers allow to expand easily the functionality of the STM32 Nucleo open development platform



1. Picture is not contractual

with a wide choice of specialized shields. The STM32 Nucleo board does not require any separate probe as it integrates the ST-LINK/V2-1 debugger and programmer. The STM32 Nucleo board comes with the STM32 comprehensive software HAL library together with various packaged software examples, as well as direct access to the ARM® mbed™ online resources at <http://mbed.org>.

Table 1. Device summary

Reference	Part number
NUCLEO-XXXXRX	NUCLEO-F030R8, NUCLEO-F070RB, NUCLEO-F072RB, NUCLEO-F091RC, NUCLEO-F103RB, NUCLEO-F302R8, NUCLEO-F303RE, NUCLEO-F334R8, NUCLEO-F401RE, NUCLEO-F410RB, NUCLEO-F411RE, NUCLEO-F446RE, NUCLEO-L053R8, NUCLEO-L073RZ, NUCLEO-L152RE, NUCLEO-L452RE, NUCLEO-L476RG.

Anexo 4

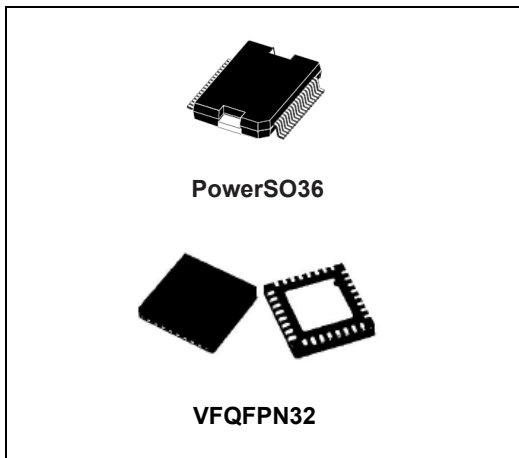
En las páginas siguientes puede encontrarse un extracto de las características definitivas del integrado L6230 obtenidas desde la hoja de características (*datasheet*), descargada de la página web del fabricante STMicroelectronics.

Han sido incluidas las páginas en las que figuran las características definitivas del dispositivo, no obstante, la documentación y hoja de características completa cuenta con un número de páginas extenso donde se detalla de forma pormenorizada cada una de las características de este elemento.

Si se desea consultar la documentación de forma completa, se recomienda acceder a través del portal web del fabricante *www.st.com*.

DMOS driver for three-phase brushless DC motor

Datasheet - production data



Description

The L6230 is a DMOS fully integrated three-phase motor driver with overcurrent protection, optimized for FOC application thanks to the independent current senses.

Realized in BCDmultipower technology, the device combines isolated DMOS power transistors with CMOS and bipolar circuits on the same chip.

An uncommitted comparator with open-drain output is available.

Available in PowerSO36 and VFQFPN32 5 x 5 packages the L6230 device features non-dissipative overcurrent protection on the high-side power MOSFETs and thermal shutdown.

Features

- Operating supply voltage from 8 to 52 V
- 2.8 A output peak current (1.4 A RMS)
- $R_{DS(on)}$ 0.73 Ω typ. value at $T_J = 25\text{ }^\circ\text{C}$
- Integrated fast freewheeling diodes
- Operating frequency up to 100 kHz
- Non-dissipative overcurrent detection and protection
- Cross conduction protection
- Diagnostic output
- Uncommitted comparator
- Thermal shutdown
- Undervoltage lockout

Application

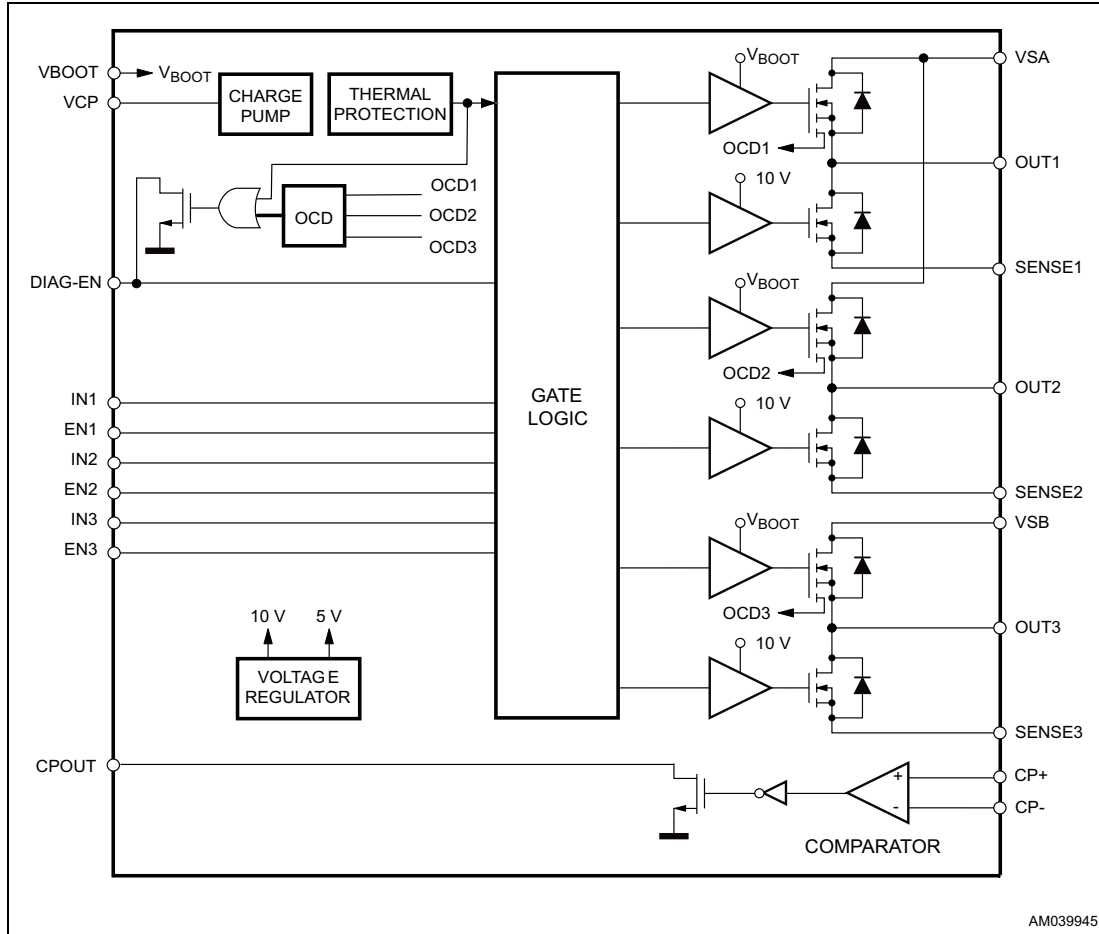
- BLDC motor driving
- Sinusoidal / six-step driving
- Field oriented control driving system

Table 1. Device summary

Order codes	Package	Packaging
L6230PD	PowerSO36	Tube
L6230PDTR		Tape and reel
L6230Q	VFQFPN32	Tube
L6230QTR		Tape and reel

1 Block diagram

Figure 1. Block diagram



2 Electrical data

2.1 Absolute maximum ratings

Table 2. Absolute maximum ratings

Symbol	Parameter	Parameter	Value	Unit
V_S	Supply voltage	$V_{SA} = V_{SB} = V_S$	60	V
V_{OD}	Differential voltage between: V_{SA} , OUT1, OUT2, SENSE1, SENSE2 and V_{SB} , OUT3, SENSE3	$V_{SA} = V_{SB} = V_S = 60\text{ V}$; $V_{SENSEx} = \text{GND}$	60	V
V_{BOOT}	Bootstrap peak voltage	$V_{SA} = V_{SB} = V_S$	$V_S + 10$	V
V_{IN} , V_{EN}	Logic inputs voltage range		-0.3 to +7	V
V_{CP-} , V_{CP+}	Voltage range at CP- and CP+ pins		-0.3 to +7	V
V_{SENSE}	Voltage range at SENSEx pins		-1 to +4	V
$I_{S(\text{peak})}$	Pulsed supply current (for each VS pin)	$V_{SA} = V_{SB} = V_S$; $T_{PULSE} < 1\text{ ms}$	3.55	A
I_S	RMS supply current (for each VS pin)	$V_{SA} = V_{SB} = V_S$	1.4	A
T_{stg} , T_{OP}	Storage and operating temperature range		-40 to 150	°C

2.2 Recommended operating conditions

Table 3. Recommended operating conditions

Symbol	Parameter	Parameter	Min	Max	Unit
V_S	Supply voltage	$V_{SA} = V_{SB} = V_S$	8	52	V
V_{OD}	Differential voltage between V_{SA} , OUT1A, OUT2A, SENSE1, SENSE2 and V_{SB} , OUT1B, OUT2B, SENSE3	$V_{SA} = V_{SB} = V_S$; $V_{SENSE1} = V_{SENSE2} = V_{SENSE3}$		52	V
V_{CP-} , V_{CP+}	Voltage range at CP- and CP+ pins		-0.1	5	V
V_{CPCM}	Common mode voltage at the comparator inputs		0	3	V
V_{SENSE}	Voltage range at pins SENSEx	pulsed $t_W < t_{rr}$	-6	6	V
		DC	-1	1	V
I_{OUT}	RMS output current			1.4	A
T_J	Operating junction temperature		-25	+125	°C
f_{sw}	Switching frequency			100	kHz

Anexo 5

En las páginas siguientes puede encontrarse un extracto de las características definitorias de la placa de expansión X-NUCLEO-IHM07M1 obtenidas desde la hoja de características (*datasheet*), descargada de la página web del fabricante STMicroelectronics.

Han sido incluidas las páginas en las que figuran las características definitorias del dispositivo, no obstante, la documentación y hoja de características completa cuenta con un número de páginas extenso donde se detalla de forma pormenorizada cada una de las características de este elemento.

Si se desea consultar la documentación de forma completa, se recomienda acceder a través del portal web del fabricante *www.st.com*.

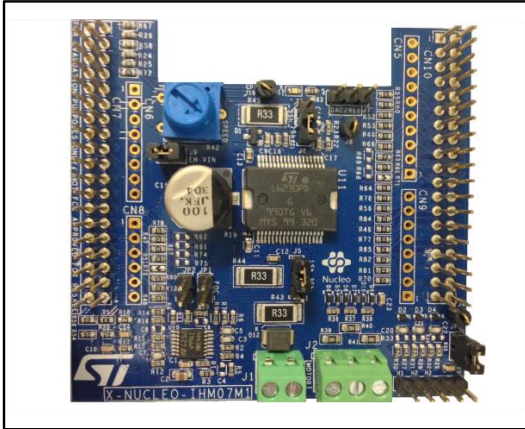
X-NUCLEO-IHM07M1



X-NUCLEO-IHM07M1

Three-phase brushless DC motor driver expansion board based on L6230 for STM32 Nucleo

Data brief



Description

The X-NUCLEO-IHM07M1 is a three-phase brushless DC motor driver expansion board based on the L6230 for STM32 Nucleo. It provides an affordable and easy-to-use solution for driving three-phase brushless DC motor in your STM32 Nucleo project. The X-NUCLEO-IHM07M1 is compatible with the ST Morpho connector and supports the addition of other boards which can be stacked with onto a single STM32 Nucleo board. The user can also mount the Arduino UNO R3 connector. The driver used on this STM32 Nucleo board is the L6230, a DMOS fully integrated driver for three-phase brushless DC motors assembled in a PowerSO-36 package (L6230PD), with overcurrent and thermal protection. The L6230 driver is optimized for six-step and FOC algorithms thanks to independent current sensing.

Features

- 3-phase driver for BLDC/PMSM motors
- Nominal operating voltage range from 8 V to 48 VDC
- 2.8 A output peak current (1.4 A_{RMS})
- Operating frequency up to 100 kHz
- Non dissipative overcurrent detection and protection
- Cross-conduction protection
- Thermal measuring and overheating protection
- Compatible with STM32 Nucleo boards
- Equipped with ST Morpho connectors
- 3-shunt and 1-shunt configurable jumpers for motor current sensing
- Hall/Encoder motor sensor connector and circuit
- Debug connector for DAC, GPIOs, etc.
- Potentiometer available for speed regulation
- User LED
- RoHS compliant

