



Facultad de Ingeniería
Escuela de Ingeniería Civil Mecatrónica

Control Predictivo de Horizonte Lejano en una Máquina Síncrona de Imanes Permanentes

Profesor Guía:
Dr. Cristián Francisco García Peñailillo

José Tomás Pérez Aedo
Curicó-Chile
2022

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



UNIVERSIDAD DE TALCA
DIRECCIÓN
SISTEMA DE BIBLIOTECAS

UNIVERSIDAD DE TALCA
SISTEMA DE BIBLIOTECAS
CAMPUS CURICO

Curicó, 2023

Control Predictivo de Horizonte Lejano en una Máquina Síncrona de Imanes Permanentes

José Tomás Pérez Aedo

Abril, 2022
José Tomás Pérez Aedo

Resumen

El siguiente documento presenta el trabajo que se llevó a cabo durante la memoria de título del alumno José Tomás Pérez, el cual, se enfoca en el desarrollo de un método de control derivado del control predictivo o MPC, particularmente, control predictivo de horizonte lejano, para mejorar el accionamiento de motores sincrónicos de imanes permanentes y mejorar la eficiencia energética en aplicaciones de conversión de potencia.

Se desarrollaron simulaciones de tres métodos de control para máquinas sincrónicas de imanes permanentes, siendo estos, control por campo orientado, control predictivo de estados finitos y control predictivo de horizonte lejano, para luego realizar una comparación en términos de distorsión armónica y frecuencia de conmutación de los semiconductores. Se generan gráficos comparativos a diferentes tiempos de muestreo entre los tres métodos de control. Luego, se implementan de forma física en un banco de pruebas experimental, utilizando una tarjeta de desarrollo LAUNCHXL-F28379D de Texas Instruments y un kit de máquinas sincrónicas del mismo fabricante, se logró la implementación del control por campo orientado y control predictivo de estados finitos, sin embargo, no se logró el funcionamiento de control predictivo de horizonte lejano, se proponen posibles razones por la que no fue posible su implementación y se compara los resultados experimentales de los dos métodos implementados de forma exitosa.

Palabras Clave

FOC, MPC, PMSM, electromovilidad.

Índice

Acrónimos	4
1 Introducción	6
1.1 Estado del Arte	7
1.2 Marco teórico	8
1.2.1 Tipos de máquinas eléctricas usadas en EV	8
1.2.2 Métodos de control para máquinas eléctricas	8
1.2.3 Tipos de control predictivo	8
1.2.4 Control predictivo de estados finitos FCS-MPC	9
1.2.5 Control predictivo de horizonte lejano LH-MPC	9
1.3 Objetivos	10
1.3.1 Objetivo General	10
1.3.2 Objetivos Específicos	10
1.4 Alcances y Limitaciones	11
1.4.1 Alcances	11
1.4.2 Limitaciones	11
1.5 Metodología	12
1.5.1 Revisión bibliográfica	12
1.5.2 Simulación FOC, FS-MPC y LH-MPC	12
1.5.3 Validación de resultados experimentales	12
1.6 Resumen de capítulos	12
2 Modelamiento de máquinas eléctricas	13
2.0.1 Ecuaciones de voltaje	13
2.0.2 Ecuaciones de corriente	14
2.0.3 Plano de referencia estacionario	14
2.0.4 Plano de referencia rotacional	15
2.0.5 Ecuaciones mecánicas	16
3 Simulación control por campo orientado	18
3.1 Diseño de controladores PI para FOC	19
3.1.1 Controlador Continuo	20
3.1.2 Controlador PI de lazo mecánico	21
3.1.3 Controlador PI de lazo eléctrico	22
3.2 Discretización del controlador PI	22
3.3 Controlador discreto con Anti Wind-Up	24
3.4 Resultados de simulación FOC	25
3.4.1 Respuesta dinámica	26
3.4.2 Estado estacionario	27
3.4.3 Distorsión armónica total	28
4 Simulación control predictivo de estados finitos	29
4.1 Modelo predictivo	30
4.2 Discretización del modelo	30
4.3 Función de costo	31
4.4 Resultados de simulación MPC	32
4.4.1 Respuesta dinámica	33
4.4.2 Estado estacionario	34
4.4.3 Distorsión armónica total versus frecuencia de conmutación promedio	35
5 Simulación control predictivo de horizonte lejano	36
5.1 Extrapolación de señales	36
5.2 Función de costo	37
5.3 Resultados de simulación LH-MPC	37

5.3.1	Respuesta dinámica	38
5.3.2	Estado estacionario	39
5.3.3	Distorsión armónica total	40
5.4	Comparación de los tres métodos	40
6	Validación experimental	42
6.1	Implementación del banco de pruebas	42
6.2	Configuración del protocolo de comunicación	43
6.3	Obtención de datos y resultados experimentales	45
7	Conclusión	49
	Anexos	65
	Referencias	65

Índice de figuras

1.2.3.1	Clasificación de tipos de control predictivo usados en electrónica de potencia.	9
1.3.0.1	Diagrama de bloques del FCS-MPC para un inversor de voltaje.	10
2.0.0.1	Circuito equivalente de un bobinado en la máquina eléctrica.	13
2.0.3.1	Representación vectorial del plano $\alpha - \beta$ proyectado sobre el plano abc de corrientes de fase.	14
2.0.4.1	Representación vectorial del plano $d - q$ proyectado sobre el plano $\alpha - \beta$	15
3.0.0.1	Diagrama de bloques de control por campo orientado.	18
3.3.0.1	Diagrama de bloques del controlador con Anti Wind-Up.	24
3.4.1.1	Resultados de respuesta dinámica de control por campo orientado.	26
3.4.2.1	Comportamiento en estado estacionario de control por campo orientado.	27
3.4.3.1	Resultados de respuesta dinámica de control por campo orientado.	28
4.0.0.1	Diagrama de control predictivo.	29
4.0.0.2	topología de un convertidor trifásico de dos niveles.	30
4.4.1.1	Resultados de respuesta dinámica de control predictivo de estados finitos.	33
4.4.2.1	Comportamiento durante estado estacionario de control predictivo de estados finitos.	34
4.4.3.1	Comparación de frecuencia de conmutación ante distintos valores de peso W en control predictivo de estados finitos.	35
5.3.1.1	Resultados de respuesta dinámica de control predictivo de horizonte lejano.	38
5.3.2.1	Comportamiento durante estado estacionario de control predictivo de horizonte lejano.	39
5.3.3.1	Comparación de frecuencia de conmutación ante distintos valores de peso W en control predictivo de horizonte lejano.	40
5.4.0.1	Comparación de THD entre los tres métodos de control propuestos.	41
6.1.0.1	Tarjeta de desarrollo de la marca Texas Instruments modelo LAUNCHXL-F28379D	42
6.1.0.2	Inversor de dos niveles de la marca Texas Instruments modelo BOOSTXL-3PHGanInv	42
6.1.0.3	PMSM de la marca Teknic modelo M-3210P-LN-04K	42
6.1.0.4	Setup experimental. Zona 1 (en amarillo): Motores acoplados y tacómetro laser. Zona 2 (en celeste): par de inversores con sensores de corriente. Zona 3 (en rojo): Computador host recibiendo datos del launchpad.	43
6.2.0.1	Ventana de configuración de hardware, en rojo, la sección de tasa de transferencia de datos.	44
6.2.0.2	Interfaz de control para el banco de pruebas	44
6.3.0.1	Osciloscopio digital Rigol, mostrando la corriente del motor controlado en celeste y la corriente del motor de carga en amarillo.	45
6.3.0.2	Corrientes de fase del motor controlado mediante MPC, con diferentes tiempos de muestreo.	46
6.3.0.3	Corrientes de fase del motor controlado mediante FOC, con diferentes tiempos de muestreo.	47
6.3.0.4	Espectro armónico de la corriente con MPC a $40\mu s$	48
6.3.0.5	Espectro armónico de la corriente con FOC a $40\mu s$	48
7.0.0.1	Captura de pantalla del osciloscopio digital Rigol, mostrando la corriente del motor controlado por MPC. Colores invertidos para mejor claridad.	50

Acrónimos

AC	Corriente alterna.
AWU	Anti-enrollamiento.
BLDC	Máquina de corriente continua sin escobillas.
DC	Corriente continua.
DTC	Control de torque directo.
DSP	Procesador digital de señales.
EMF	Fuerza electromotriz.
EV	Vehículo eléctrico.
FOC	Control por campo orientado.
FPGA	Arreglo de puertas lógicas programable.
FS-MPC	Control predictivo de estados finitos.
IGBT	Transistor bipolar de compuerta aislada.
MPC	Control predictivo basado en modelos.
NPC	Inversor multinivel con diodo anclado al neutro.
LH-MPC	Control predictivo de horizonte lejano.
PI	Proporcional e Integral.
PID	Proporcional, Integral y Derivativo.
PMSM	Máquina síncrona de imanes permanentes.
PWM	Modulación por ancho de pulso.
RPM	Revoluciones por minuto.
THD	Distorsión armónica total.

1. Introducción

En la actualidad, los vehículos que utilizan combustibles fósiles para su funcionamiento, contribuyen al incremento en las emisiones de gases contaminantes y de efecto invernadero. Por otra parte, los vehículos híbridos presentan una mejor opción frente a los vehículos tradicionales, pero estos son solamente una opción transitoria hasta que se se logre llegar al objetivo último, que son los vehículos totalmente eléctricos.

Los vehículos eléctricos, llevan años siendo desarrollados para mejorar su autonomía y capacidad de carga. Sin embargo, aún no se logra competir con los combustibles fósiles en viajes de largas distancias, por ejemplo, el Modelo S de Tesla, con un precio aproximado de \$72.000.000, cuenta con una autonomía estimada de 637 Km [1], por otro lado el Peugeot 208, con un precio aproximado de \$18.000.000, cuenta con un rendimiento promedio de 22,2 Km/L y con un estanque de 50 litros [2], se obtiene una autonomía de 1110 Km aproximadamente, casi el doble que el modelo S, y a un precio mucho menor.

Otra arista dentro los vehículos eléctricos, en inglés electric vehicle (EV), es la de los EV de transporte unipersonal, como lo son; bicicletas eléctricas, Scooters, Skates, entre otros. Estos vehículos, tienen el potencial de disminuir los atochamientos, además de ser una buena opción para personas que viajan al trabajo de forma rutinaria y que la distancia a su lugar de destino no es muy lejana.

Típicamente los vehículos eléctricos utilizan dos diseños diferentes al colocar una máquina eléctrico en ellos, en uno, el motor está conectado al eje de la rueda a través de diferenciales y/o reductores, y en el otro, el motor se construye dentro de la rueda para una tracción directa. Dentro de todos los tipos de máquinas eléctricas, con respecto al área de electro-movilidad se usan normalmente tres tipos, estos son, máquinas de corriente directa (DC), máquinas de inducción (IM) y máquinas de imanes permanentes (PM). Cada uno tiene aplicaciones dependiendo del tipo de vehículo que se quiera diseñar [3].

Este trabajo se enfocará en las máquinas de imanes permanentes, particularmente en los motores DC sin escobillas o BLDC por sus siglas en inglés “Brushless DC”, estos motores requieren un tipo de control especial para su accionamiento, deben ser controlados con corrientes alternas multifásicas, pues, al no contar con delgas, cada bobinado debe ser encendido síncronamente para generar el movimiento del eje.

Actualmente existen diversas metodologías de control para estos motores, y es en este aspecto, en el que se puede generar propuestas de control que contribuyan al mejoramiento de la eficiencia energética y lograr una mejor autonomía para los vehículos eléctricos. En particular un método que parece prometedor es el método de control predictivo de horizonte lejano, donde se utiliza el modelo matemático del motor para predecir el comportamiento de las variables del motor en varios pasos de tiempo en el futuro, y se utiliza esta información para calcular el mejor rumbo de acción.

1.1. Estado del Arte

Dentro de la literatura revisada, se encuentran artículos acerca de control predictivo clásico para máquinas BLDC, desarrollando diferentes metodologías y enfocándose en distintos parámetros del sistema como son; el control de torque directo (DTC); y control predictivo de corriente (PCC) [4]. Además, métodos de control basados en transformaciones de los planos de referencia de las corrientes como el método de control por campo orientado (FOC) [5].

Ambas propuestas pueden ser aprovechadas para la implementación de un control predictivo de estados finitos, permitiendo que el código generado para controlar el convertidor de potencia sea menos complejo, esto es fundamental para que su ejecución se rápida, y los cálculos se realicen en el menor tiempo posible.

Por otro lado, se proponen mejoras a los métodos de control clásicos para reducir el Ripple al aplicar DTC, en diferentes topologías de convertidores de potencia [6].

También se habla de una novedosa implementación en tiempo real de control predictivo lineal basado en métodos de control predictivo explícito sin regiones o “Region-less explicit MPC”, que propone una reducción en el uso de memoria permitiendo el uso de LH-MPC, además, se presentan arquitecturas para su implementación en FPGA y se valida con aplicaciones de convertidores buck-boost y modular multinivel [7].

Otros artículos exploran el uso de decodificación de esfera, para optimizar el cálculo de las posibles ramificaciones en los tiempos discretos futuros del control predictivo de horizonte lejano y se evalúa su rendimiento para ser implementada en un FPGA de bajo costo [8], además, el uso de la decodificación de esfera para mejorar el rendimiento de los inversores de voltaje de dos niveles y de cuatro piernas conectados a la red eléctrica [9].

Se revisó un artículo que analiza la robustez del control predictivo de horizonte lejano al ser aplicado a motores de imanes permanentes, y se propone la adición de un componente integrador dentro del problema de optimizador sin agregar complejidad ni perjudicar la capacidad computacional [10].

Se revisó varios documentos ([11], [12], [13]) que comparan la distorsión armónica total, de los diferentes métodos de control (FOC, FS-MPC, LH-MPC) versus la frecuencia de conmutación en cada caso, además, de comparar los propios métodos de control predictivo con diferentes valores de peso que alteran la frecuencia de conmutación.

1.2. Marco teórico

1.2.1. Tipos de máquinas eléctricas usadas en EV

- **Máquinas DC:** Se usan por su facilidad de control, que es un factor importante a considerar en su aplicación para vehículos eléctricos, sin embargo, son poco confiables debido a su funcionamiento dependiente de contactos mecánicos entre las delgas y las escobillas, este roce constante produce desgaste en el motor, requiriendo de mantención constante. Esto se debe evitar lo mas posible y su uso se ha dejado de lado en favor de los otros tipos de motores eléctricos.
- **Máquinas de inducción:** Este tipo de máquinas son mas utilizados en los EV, gracias a su bajo costo, robustez, y alta eficiencia en la zona de potencia constante. Sin embargo estos motores poseen algunas desventajas, como su baja eficiencia a bajas revoluciones, baja densidad de torque, se necesita un tipo de control de vectores para lograr un desempeño deseado, debido a su frecuencia y voltaje variable.
- **Máquinas de imanes permanentes:** Este tipo de máquinas son mas atractivos en todo tipo de aplicaciones para EV, por sus varias ventajas, como, alta eficiencia energética, alta densidad de torque y gran robustez. Sin embargo, tienen un costo mas elevado que otro tipo de motores debido a los materiales necesarios para la construcción de los imanes permanentes.

Este tipo de motores se puede dividir en dos categorías dependiendo de la forma de onda de su fuerza contra electromotriz (back EMF), estas son, motores de imanes permanentes de onda sinusoidal, o descritos de otra forma como motores de imanes permanentes sincrónicos (PMSM) y por otro lado, los motores DC sin escobillas (BLDC) que tiene una forma de onda trapezoidal para su fuerza contra electromotriz.

1.2.2. Métodos de control para máquinas eléctricas

Control directo de torque (DTC): Este tipo de control no requiere de un modulador, ni de un lazo de control PI de corriente, estas características hacen que su implementación en un sistema sea fácil y de rápida respuesta dinámica [14]. La selección de vectores de voltaje se hace dependiendo de la diferencia entre la referencia y el flujo magnético entre el torque real y el estator.

Para este tipo de control se debe tener una tabla de valores previamente declarados para la salida del sistema. Algunas desventajas de este método de control es que posee un alto ripple de torque y una respuesta transiente lenta durante el inicio.

Control de campo orientado (FOC): Este método permite el control independiente del torque y el flujo magnético, al desacoplar la corriente del estator en sus componentes ortogonales. Sin embargo es muy susceptible a la variación del flujo magnético que es afectado por variaciones en los parámetros, por eso es importante definir valores precisos para alcanzar el rendimiento esperado [15].

Control predictivo basado en modelos (MPC): La principal característica del control predictivo es el uso de un modelo matemático para calcular el comportamiento futuro de las variables controladas en un sistema. esta información es utilizada por el controlador para determinar el mejor curso de acción de acuerdo a un criterio de optimización predefinido. La figura 1.2.3.1 muestra una posible clasificación para los tipos de control predictivo.

1.2.3. Tipos de control predictivo

El control predictivo basado en modelos ha tenido cerca de cuatro décadas de desarrollo continuo, y es considerado uno de los avances mas importantes en el control de procesos [16]. El control predictivo ofrece muchas ventajas, en particular puede ser utilizado en una amplia variedad de procesos, siendo de

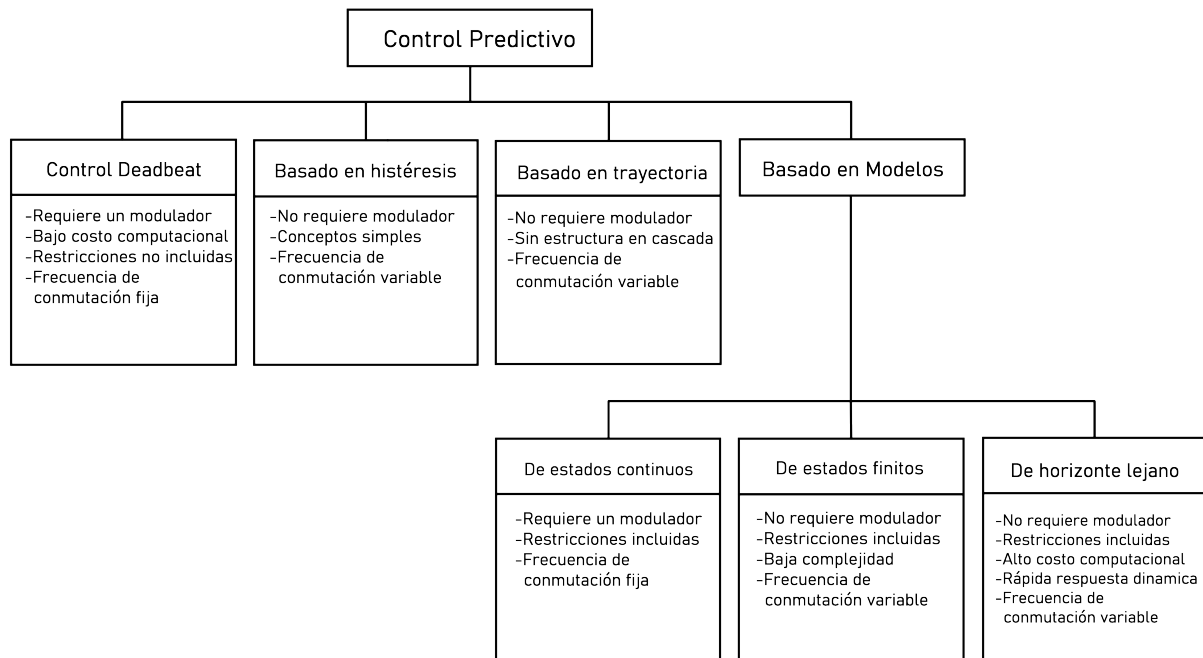


Fig. 1.2.3.1: Clasificación de tipos de control predictivo usados en electrónica de potencia.

fácil implementación en sistemas multivariables. A pesar de tener un mayor costo de cómputo que otros métodos de control, como PID o control lineal, esto no genera un problema gracias al creciente poder de procesamiento de los dispositivos digitales, como las FPGA y DSP que han sido ampliamente utilizados en el área de conversión de potencia.

1.2.4. Control predictivo de estados finitos FCS-MPC

Al utilizar el control predictivo de estados finitos en el ámbito de electrónica de potencia, se tiene una ventaja particular, pues los convertidores de potencia tienen una cantidad limitada de estados de conmutación, por lo que se puede resolver fácilmente el cálculo de la optimización del modelo. Típicamente se usan modelos discretos para predecir el comportamiento del sistema frente a cada secuencia de actuación posible hasta el horizonte de predicción. El estado de conmutación que minimiza una función de costo predefinida es elegida y aplicada en el instante de muestreo siguiente. La ventaja principal del FCS-MPC se encuentra en la aplicación directa de la acción de control al convertidor de potencia, sin requerir una etapa de modulación [17].

1.2.5. Control predictivo de horizonte lejano LH-MPC

Como ya se ha mencionado anteriormente, el FCS-MPC es utilizado predominantemente para aplicaciones de conversión de potencia, debido a su versatilidad y rápida respuesta dinámica. Sin embargo, cuando se adoptan horizontes lejanos para predecir las trayectorias de las variables de estado durante múltiples tiempos de muestreo, se produce un mejoramiento substancial frente a predicciones de corto tiempo, al reducir la distorsión armónica de la corriente en el sistema. La aplicación de horizontes lejanos son potencialmente beneficiosos para plantas de ordenes elevados, como los convertidores con filtros LC.

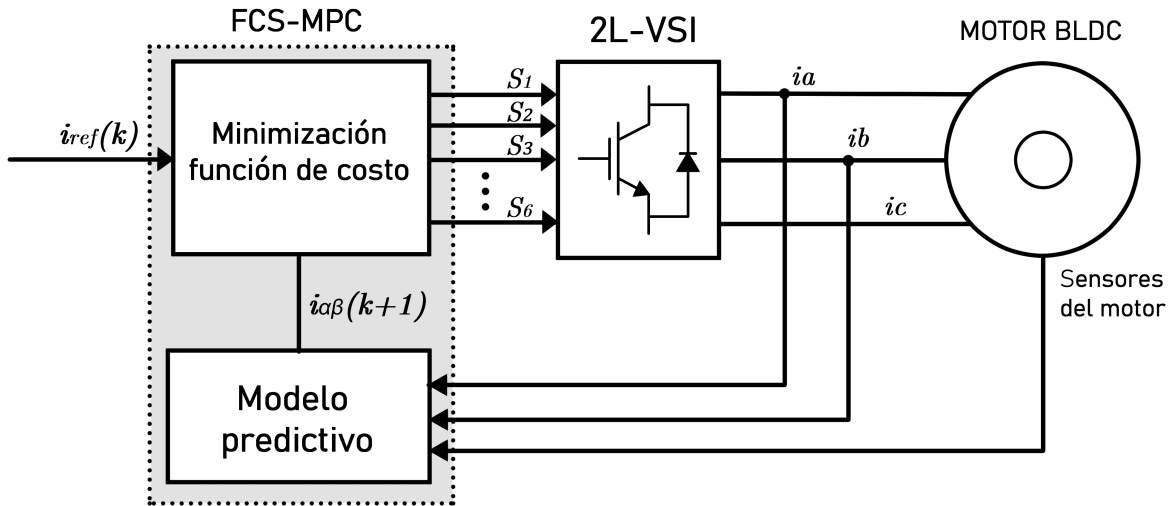


Fig. 1.3.0.1: Diagrama de bloques del FCS-MPC para un inversor de voltaje.

1.3. Objetivos

1.3.1. Objetivo General

Reducir la distorsión armónica total en motores síncronos utilizando control predictivo de horizonte lejano.

1.3.2. Objetivos Específicos

- Realizar una revisión bibliográfica respecto a tipos de control predictivo y maquina eléctricas.
- Comparar la distorsión armónica total entre LH-MPC, FCS-MPC y FOC.
- Comparar el rendimiento estacionario y dinámico entre LH-MPC, FCS-MPC y FOC.

1.4. Alcances y Limitaciones

1.4.1. Alcances

Este trabajo contempla la implementación y simulación de control predictivo de estados finitos, control predictivo de horizonte lejano y control de campo orientado.

Se utilizará un convertidor de dos niveles y una plataforma de control comercial.

Se utilizará kit de desarrollo que incluye dos motores y un acople. Se simulará el efecto de tracción en el motor estudiado, utilizando un motor acoplado como carga.

1.4.2. Limitaciones

El trabajo a desarrollar no contempla la implementación de otras estrategias de control, como control de torque directo, o control predictivo de torque, etc.

No se implementarán en otras topologías de inversores, como NPC o inversor de tres niveles.

Se utilizará una fuente de voltaje controlado para reemplazar la alimentación por baterías que típicamente se encuentran en un vehículo eléctrico.

1.5. Metodología

1.5.1. Revisión bibliográfica

El proyecto contempla una etapa revisión bibliográfica, donde se recopilarán documentos relacionados a los tipos de máquinas eléctricas utilizados en EV, la aplicación de control predictivo en convertidores de potencia, y los distintos métodos de optimización de los modelos.

1.5.2. Simulación FOC, FS-MPC y LH-MPC

Otra etapa es la de simulación, donde se desarrollará un modelo del sistema de control, el convertidor de potencia y el motor BLDC, para poder hacer pruebas de los distintos métodos de control, como control predictivo de estados finitos, y de horizonte lejano. Estas simulaciones se llevaran a cabo en el ambiente de simulación Simulink del software MATLAB.

1.5.3. Validación de resultados experimentales

La etapa final consiste en la construcción de un banco de pruebas real, donde se podrán hacer pruebas experimentales con motores físicos y una tarjeta programable.

1.6. Resumen de capítulos

El trabajo a realizar contempla los siguientes capítulos

1. Introducción

Se hablará acerca del enfoque del proyecto, descripción de conceptos, marco teórico, objetivos, estado del arte, alcances, y la metodología a usar.

2. Modelamiento de máquinas eléctricas

Se explica la teoría y los procedimientos matemáticos necesarios para obtener el modelo continuo y discreto de la máquina eléctrica estudiada.

3. Simulación de control por campo orientado

Se explica la teoría detrás del método de control por campo orientado y se grafican los resultados de simulación.

4. Simulación de control predictivo de estados finitos

Se explica la teoría detrás del método de control predictivo de estados finitos y se grafican los resultados de simulación.

5. Simulación de control predictivo de horizonte lejano

Se explica la teoría detrás del método de control predictivo de estados finitos, se grafican los resultados de simulación y se comparan los tres métodos.

6. Validación experimental

Se indican los materiales e instrumentos usados en la creación del banco experimental y se desarrollan los experimentos de control.

2. Modelamiento de máquinas eléctricas

Se comenzará definiendo el modelo matemático de una máquina eléctrica de imanes permanentes, para posteriormente ser usado en los métodos de control propuestos. El circuito equivalente para una fase de la máquina eléctrica se puede representar como se ilustra en la figura 2.0.0.1.

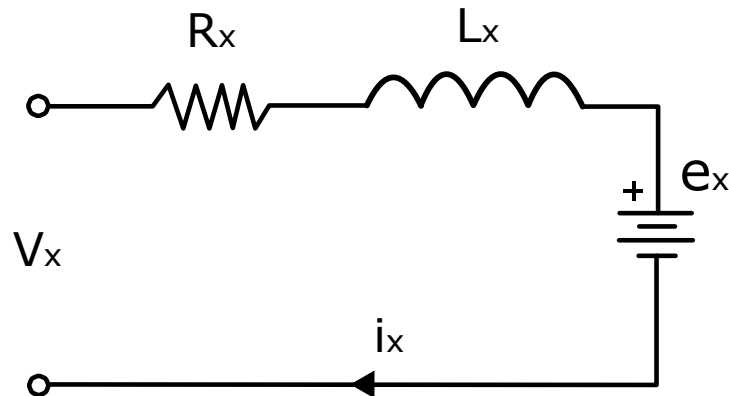


Fig. 2.0.0.1: Circuito equivalente de un bobinado en la máquina eléctrica.

2.0.1. Ecuaciones de voltaje

Las ecuaciones que representan los voltajes de las distintas fases son:

$$V_a = Ri_a + L \frac{di_a}{dt} + e_a \quad (1)$$

$$V_b = Ri_b + L \frac{di_b}{dt} + e_b \quad (2)$$

$$V_c = Ri_c + L \frac{di_c}{dt} + e_c \quad (3)$$

En el caso de una máquina de imanes permanentes, la fuerza electromotriz y la variación de la corriente en cualquier fase, corresponde a la variación del flujo magnético del estator, por lo tanto las ecuaciones se representan como:

$$V_f = Ri_f + \frac{d\Phi_f}{dt} \quad (4)$$

y a su vez, el flujo magnético es:

$$\Phi_f = Li_f + \lambda_{PM} \quad (5)$$

Donde λ_{PM} es el flujo magnético inducido por los imanes permanentes del rotor y depende del ángulo del estator con respecto al rotor. Por esto, al derivar el término de la ecuación 5 aparece un término de velocidad angular de las corrientes de fase ω_e como muestra la ecuación 6:

$$V_f = Ri_a + L \frac{di_f}{dt} + j\omega_e \lambda_{PM} \quad (6)$$

2.0.2. Ecuaciones de corriente

El modelo de la máquina de imanes permanentes consisten de tres bobinados en el estator típicamente conectados en configuración estrella, con un ángulo de 120° entre ellos. Debido a la geometría de la bobinas, las corrientes que pasan por ellas quedan como:

$$i_a = I_s \cos(\omega_e t) \quad (7)$$

$$i_b = I_s \cos(\omega_e t + \theta - 2\pi/3) \quad (8)$$

$$i_c = I_s \cos(\omega_e t + \theta + 2\pi/3) \quad (9)$$

Donde I_s y θ , son la amplitud y el ángulo inicial de las corrientes de fase, respectivamente.

2.0.3. Plano de referencia estacionario

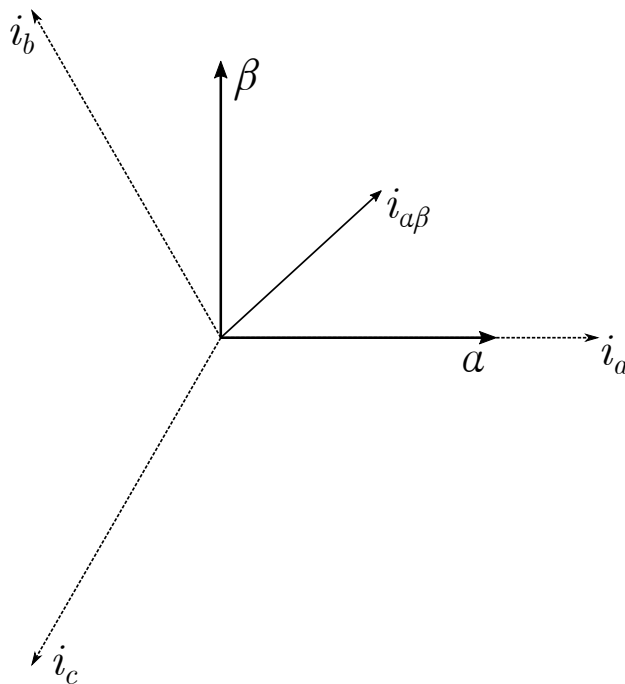


Fig. 2.0.3.1: Representación vectorial del plano $\alpha - \beta$ proyectado sobre el plano abc de corrientes de fase.

Controlar la corriente en una máquina con corrientes de 3 fases es una tarea compleja por si sola, por esto se recurre a transformaciones matemáticas, que permitan manipular la corriente de forma sencilla, una de estas transformaciones es la transformada de Clark. Esta convierte las componentes del dominio del tiempo de un sistema de tres fases, en dos componentes de un marco estacionario ortogonal denominado plano Alfa-Beta (Fig. 2.0.3.1). La transformada de Clark se puede representar matricialmente como:

$$\begin{bmatrix} \alpha \\ \beta \\ z_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (10)$$

Donde z_0 es la componente de secuencia cero y para un sistema trifásico balanceado, su valor es cero. Aplicando esta transformada a las corrientes de fase, estas quedan como:

$$\begin{bmatrix} i_\alpha \\ i_\beta \\ i_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (11)$$

Del mismo modo, al aplicar la transformada de Clark a la ecuación 6, Los voltajes en el plano $\alpha - \beta$, son:

$$V_\alpha = Ri_\alpha + L \frac{di_\alpha}{dt} - j\omega_e \lambda_{PM} \sin(\theta_e) \quad (12)$$

$$V_\beta = Ri_\beta + L \frac{di_\beta}{dt} + j\omega_e \lambda_{PM} \cos(\theta_e) \quad (13)$$

Donde θ_e es el ángulo eléctrico del estator, con respecto al plano de referencia original a,b,c.

2.0.4. Plano de referencia rotacional

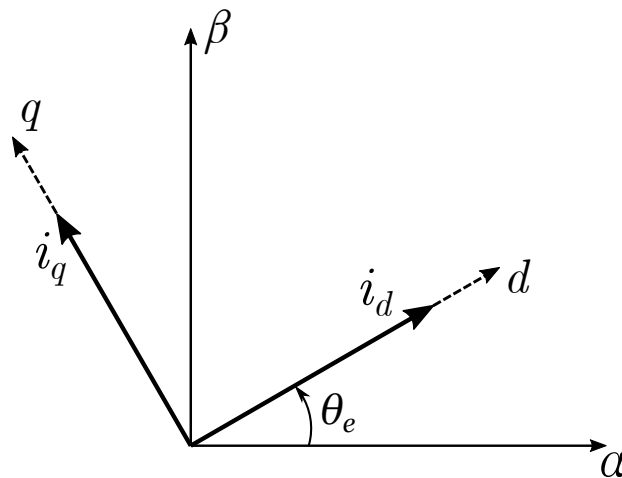


Fig. 2.0.4.1: Representación vectorial del plano $d - q$ proyectado sobre el plano $\alpha - \beta$.

La transformada de Park convierte las dos componentes del marco estacionario Alfa-Beta a un marco de referencia rotatorio ortogonal denominado d-q, relacionados a través del ángulo de rotación del campo eléctrico del rotor (θ_e). Implementar estas dos transformadas de manera consecutiva simplifica los cálculos, al convertir la forma de onda de las magnitudes sinusoidales en señales continuas. La ecuación matricial que describe esta transformación es:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos(\theta_e) & \sin(\theta_e) \\ -\sin(\theta_e) & \cos(\theta_e) \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (14)$$

Alineando el eje directo (i_d) del nuevo plano con el vector de flujo producido por los imanes permanentes, podemos despreciar este término, pues, al rotar con el mismo ángulo que el plano de referencia su componente se hace 0 y la componente en el eje de cuadratura (i_q) puede ser controlada directamente.

Al aplicar las transformaciones Clark y Park a los voltajes de las fases abc, podemos obtener los voltajes en el marco de referencia d-q.

$$\begin{bmatrix} V_d \\ V_q \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta_e) & \sin(\theta_e) \\ -\sin(\theta_e) & \cos(\theta_e) \end{bmatrix} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \quad (15)$$

De esta transformación resultan las ecuaciones 16 y 17

$$V_d = Ri_d + L \frac{di_d}{dt} - \omega_e Li_q \quad (16)$$

$$V_q = Ri_q + L \frac{di_q}{dt} + \omega_e Li_d + j\omega_e \lambda_{PM} \quad (17)$$

2.0.5. Ecuaciones mecánicas

Utilizando las ecuaciones de corriente y voltaje en el plano $d - q$ podemos definir el torque eléctrico en una máquina de imanes permanentes como:

$$T_e = p(\Phi_d i_q - \Phi_q i_d) \quad (18)$$

Donde p es el número de pares de polos de la máquina y la ecuación de flujo (5) está representada en el plano $d - q$.

Al hacer una multiplicación término a término, la ecuación anterior queda:

$$T_e = p((Li_d i_q + \lambda_{PM} i_q) - (Li_q i_d + i_d \lambda_{PM})) \quad (19)$$

Como se definió anteriormente, al alinear el ángulo de rotación eléctrico θ_e con el plano $d - q$ la componente de la corriente i_d se hace cero, por lo tanto el torque eléctrico puede ser controlado con la variación de la corriente i_q exclusivamente.

En una máquina síncrona, la relación de la velocidad angular eléctrica ω_e y la velocidad angular del eje mecánico ω_m es proporcional a la cantidad de pares de polos (p) del estator.

$$\omega_e = p\omega_m \quad (20)$$

La ecuación de torque mecánico de una máquina de imanes permanentes queda descrita como:

$$T_m = J\omega_m + B \frac{d\omega_m}{dt} + T_L \quad (21)$$

Donde J es la inercia del eje, B es coeficiente de fricción viscosa, y T_L es el torque de carga aplicado a la máquina.

Luego, las ecuaciones 22, 23 y 24 representan un re-arreglo de las ecuaciones de voltaje en el plano d-q y la ecuación mecánica en un tiempo continuo, donde la derivada de la corriente y la derivada de la velocidad angular es el factor despejado.

$$\frac{di_d(t)}{dt} = \frac{V_d(t)}{L} - \frac{R}{L}i_d(t) - \omega_e(t)i_q(t) \quad (22)$$

$$\frac{di_q(t)}{dt} = \frac{V_q(t)}{L} - \frac{R}{L}i_q(t) + \omega_e(t)i_d(t) + j\omega_e(t)\lambda_{PM} \quad (23)$$

$$\frac{d\omega_m(t)}{dt} = \frac{T_m(t)}{B} - \frac{J}{B}i_q(t) - T_L(t) \quad (24)$$

3. Simulación control por campo orientado

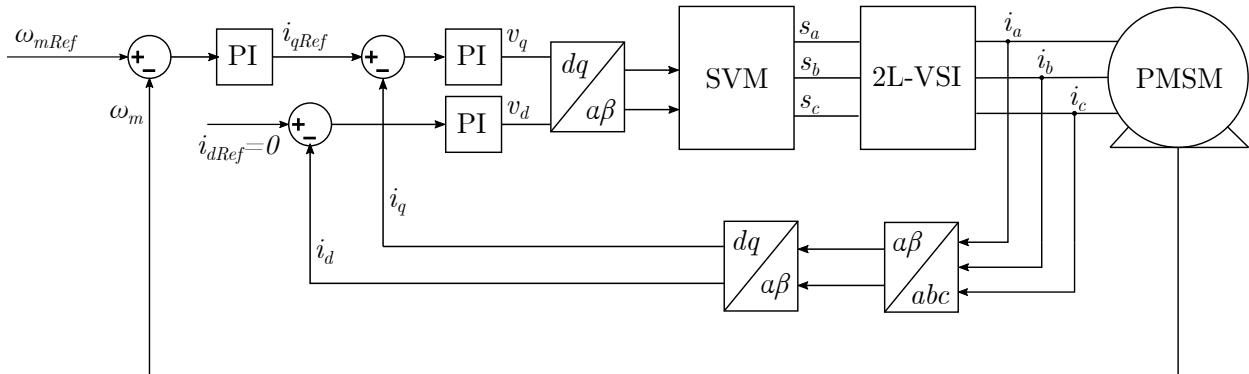


Fig. 3.0.0.1: Diagrama de bloques de control por campo orientado.

El control por campo orientado en una forma clásica de control para máquinas eléctricas, el diagrama de bloques que representa este tipo de control se puede observar en la figura 3.0.0.1.

Para simular los distintos métodos de control, se utilizó el ambiente de Simulink en Matlab. Algunos componentes se han tomado de la biblioteca Simscape, Powersim y el toolbox de control de motores de Simulink, otros se han creado como funciones de Matlab o como bloques individuales. Los parámetros del motor corresponden a los del modelo Teknic M-2310P-LN-04K, los cuales se pueden apreciar en la tabla 3.0.0.1

Parámetro	Valor
Resistencia de estator	0.32 Ω
Inductancia de estator	0.21 mH
Corriente máxima	7.1 A
Voltaje máximo	60 V
Inercia del eje	7.06 μ Kg m^2
Torque nominal	0.27 Nm
Velocidad nominal	4000 RPM
Cuentas del encoder	4000
Constante de torque	0.038 Nm/A
Constante eléctrica	4.64 V/KRPM

Tabla 3.0.0.1: Parametros del motor Teknic M-2310P-LN-04K

Como se describió en la sección 2 las corrientes sinusoidales de las fases del motor pueden describirse como corrientes continuas aplicando las transformadas matemáticas de Clarke y Park. Las corrientes de fase trifásicas del motor i_a , i_b e i_c se miden y se transforman en un sistema de coordenadas 2D con dos valores, i_d e i_q . El conocimiento de la posición física del rotor y, por lo tanto, del ángulo eléctrico del rotor es esencial para realizar las transformaciones de Clark y Park y sus correspondientes transformadas inversas. La figura 3.0.0.1 muestra el diagrama de bloques del control de campo orientado.

Este algoritmo de control consta de dos controladores PI para calcular las señales de control v_d y v_q que se debe aplicar al VSI para regular las corrientes de la máquina en el marco de referencia $d - q$. La ganancia proporcional (P) se utiliza para controlar la desviación del valor de referencia mientras que la ganancia integral (I) se utiliza para reducir los errores en estado estacionario.

En cada iteración del lazo de control se toman mediciones para determinar el error de la corriente con respecto a los valores de referencia. Uno de los controladores regula la corriente del eje i_q y el otro la corriente

directa del eje i_d del estator. Las salidas de los controladores PI, v_d y v_q , se transforman de nuevo en variables trifásicas mediante la transformación inversa de Park y Clarke y se alimentan al módulo PWM, este último, determina las señales de activación para cada uno de los semiconductores del inversor.

3.1. Diseño de controladores PI para FOC

Para diseñar los controladores PI de corriente del control por campo orientado, se utilizó el modelo eléctrico del motor visto en la sección 2 aplicando la transformada de Laplace a las ecuaciones 22, 23 y 24. Para obtener una función de transferencia para las ecuaciones eléctricas y mecánicas del sistema. Estas ecuaciones se pueden representar en el plano de Laplace como:

$$V_d(s) = (sL + R) i_d(s) - \omega_e L i_q(s) \quad (25)$$

$$V_q(s) = (sL + R) i_q(s) + \omega_e (L i_d(s) + \lambda_{PM}) \quad (26)$$

$$T_m(s) = (sB + J) \omega_m(s) + T_L \quad (27)$$

Como se puede observar en las ecuaciones de voltaje, ambas están acopladas debido a la fuerza electromagnética inducida por las corrientes del eje perpendicular respectivo, por esto, se agrega un termino en forma de perturbación para desacoplar ambas ecuaciones. De esta forma las ecuaciones quedan:

$$V_d(s) = (sL + R) i_d(s) \quad (28)$$

$$V_q(s) = (sL + R) i_q(s) + \omega_e \lambda_{PM} \quad (29)$$

Ahora se puede despejar las ecuaciones para dejarlas en forma de función de transferencia, quedando las ecuaciones eléctricas como:

$$i_d(s) = V_d \left(\frac{1}{sL + R} \right) \quad (30)$$

$$i_q(s) = (V_q - \omega_e \lambda_{PM}) \left(\frac{1}{sL + R} \right) \quad (31)$$

Y la ecuación mecánica como:

$$\omega_m(s) = (T_m - T_L) \left(\frac{1}{sJ + B} \right) \quad (32)$$

3.1.1. Controlador Continuo

La ecuación general de un controlador PI en el dominio del tiempo, es:

$$u(t) = K_p \cdot e(t) + K_i \int e(t) dt \quad (33)$$

Al aplicar la transformada de Laplace, la ecuación anterior en el dominio de la frecuencia, queda:

$$u(s) = K_p + \frac{K_i}{s} \quad (34)$$

Donde K_p es la ganancia proporcional, y K_i es la ganancia integral. Además, se puede representar como función de transferencia del controlador C de la forma:

$$C(s) = K_p \left(\frac{s + \tau_i}{s} \right) \quad (35)$$

Donde $\tau_i = \frac{K_i}{K_p}$. Luego, la función de transferencia generalizada para una planta de primer orden es:

$$G(s) = \frac{k_0}{s\tau_0 + 1} \quad (36)$$

Aplicando el controlador en lazo cerrado, la función de transferencia total queda como:

$$T(s) = \frac{C(s) \cdot G(s)}{1 + C(s) \cdot G(s)} = \frac{\frac{K_o K_p}{\tau_o T_i} (T_i \cdot s + 1)}{s^2 + \left(\frac{1 + K_o K_p}{\tau_o} \right) s + \frac{K_o K_p}{\tau_o T_i}} \quad (37)$$

Donde $T_i = \frac{1}{\tau_i}$. La ecuación anterior se puede re-parametrizar para asemejarse a la función de transferencia generalizada para una planta de segundo orden, de la siguiente forma:

$$s^2 + 2\xi\omega_n \cdot s + \omega_n^2 = s^2 + \left(\frac{1 + K_o K_p}{\tau_o} \right) s + \frac{K_o K_p}{\tau_o T_i} \quad (38)$$

De esta forma se puede seleccionar la ganancia proporcional K_p y el tiempo integral T_i al modificar el factor de amortiguamiento ξ y la frecuencia natural, o ancho de banda del sistema ω_n , ya que:

$$\omega_n^2 = \frac{K_o K_p}{\tau_o T_i} \quad (39)$$

$$\xi = \frac{1 + K_o K_p}{2\omega_n \tau_o} \quad (40)$$

despejando los parámetros de interés K_p y T_i de las ecuaciones anteriores se tiene:

$$T_i = \frac{2\xi\omega_n\tau_o - 1}{\omega_n^2\tau_o} = \frac{2\xi}{\omega_n} - \frac{1}{\omega_n^2\tau_o} \quad (41)$$

$$K_p = \frac{2\xi\omega_n\tau_o - 1}{K_o} = \frac{\omega_n^2\tau_o}{K_o} T_i \quad (42)$$

3.1.2. Controlador PI de lazo mecánico

Aplicando la técnica anterior al modelo mecánico de la máquina de imanes permanentes, se puede diseñar un controlador para el lazo de velocidad, dado por la ecuación 32. Por lo tanto, la función de transferencia queda como:

$$G_\omega(s) = \frac{\omega_m(s)}{i_{dq}(s)} = \frac{k_T}{sJ + B} \quad (43)$$

Luego, al re-parametrizar la ecuación para que quede como la función de transferencia generalizada:

$$k_m = \frac{k_T}{B} \quad (44)$$

$$\tau_m = \frac{J}{B} \quad (45)$$

Reemplazando los valores de cada parámetro por los de la máquina, según la tabla 3.0.0.1:

$$k_m = 3982 \quad \text{y} \quad \tau_m = 1.82$$

Con esto, la frecuencia natural del sistema mecánico es de $0.087Hz$, por lo que se escoge un ancho de banda por encima de este valor, para que la respuesta dinámica sea más rápida, el ancho de banda y el coeficiente de amortiguación seleccionados para el lazo mecánico son:

$$\omega_{BWm} = 2.6Hz \quad \text{y} \quad \xi_m = 1.7$$

Sabiendo que el Ancho de banda es proporcional a la frecuencia natural por un factor μ de la forma:

$$\omega_{BW} = \mu \cdot \omega_n \quad (46)$$

y que el factor μ es:

$$\mu = \sqrt{1 + \xi^2 + \sqrt{4\xi^4 + 4\xi^2 + 2}} \quad (47)$$

Se puede calcular la nueva frecuencia ajustada para obtener los valores de las ganancias proporcional e integral para en controlador PI. Luego, reemplazando el valor seleccionado para el coeficiente de amortiguación y la nueva frecuencia natural en las ecuaciones 41 y 42 se obtiene:

$$K_{pm} = 0.01 \quad \text{y} \quad T_{im} = 0.66$$

3.1.3. Controlador PI de lazo eléctrico

De la misma forma que en el lazo de velocidad, se puede obtener los parámetros de diseño para el controlador del lazo eléctrico de la máquina.

$$G_e(s) = \frac{i_{dq}(s)}{V_{dq}(s)} = \frac{1}{sL + R} \quad (48)$$

Al re-parametrizar la ecuación para que quede como la función de transferencia generalizada:

$$k_e = \frac{1}{R} \quad (49)$$

$$\tau_e = \frac{L}{R} \quad (50)$$

Reemplazando los valores de la resistencia e inductancia, según la tabla 3.0.0.1:

$$k_e = 3.13 \quad \text{y} \quad \tau_e = 6.25 \cdot 10^{-4}$$

Para el sistema eléctrico la frecuencia natural es de $95.26Hz$ en este caso, los valores de ancho de banda y coeficiente de amortiguación son:

$$\omega_{BW_m} = 190Hz \quad \text{y} \quad \xi_m = 0.707$$

Luego, calculando el nuevo valor de ω_n a partir de la ecuación 46 y 47 para posteriormente calcular las ganancias según las ecuaciones 42 y 41 se obtiene:

$$K_{pe} = 0.16 \quad \text{y} \quad T_{ie} = 6.99 \cdot 10^{-4}$$

3.2. Discretización del controlador PI

Los controladores de plantas se implementan en dispositivos digitales, como microprocesadores y PLC, estos dispositivos operan de forma discreta, a una frecuencia definida por los ciclos de reloj del procesador. Es por esto no se puede aplicar directamente el controlador diseñado en un tiempo continuo, y se debe aplicar una discretización para que puedan funcionar adecuadamente en estos dispositivos. Al considerar la siguiente estructura de un controlador PI continuo:

$$C(s) = \left(\frac{K_p + K_p T_i s}{T_i s} \right) \quad (51)$$

El controlador PI discreto tiene la siguiente estructura:

$$C(z) = \frac{K_{N0} + K_{N1} \cdot z^{-1}}{1 - z^{-1}} \quad (52)$$

Por lo que K_{N0} y K_{N1} son las nuevas variables a determinar. Algunos tipos de discretización conocidos son la discretización de Euler, y la discretización de Tustin. El controlador diseñado en este trabajo utiliza la discretización de Tustin, que considera la siguiente aproximación de la integral:

$$\frac{1}{s} \approx \frac{T_s}{2} \frac{1 + z^{-1}}{1 - z^{-1}}$$

Por lo tanto, de la función inversa la derivada se puede aproximar como:

$$s \approx \frac{2}{T_s} \frac{1 - z^{-1}}{1 + z^{-1}}$$

Luego, reemplazando s en la ecuación 51 se obtiene:

$$C(z) = \frac{\left(K_p + K_p T_i \cdot \frac{2}{T_s} \frac{1 - z^{-1}}{1 + z^{-1}} \right)}{T_i \cdot \frac{2}{T_s} \frac{1 - z^{-1}}{1 + z^{-1}}} \quad (53)$$

Al desarrollar la expresión esta queda de la forma:

$$\frac{(K_p T_s (1 + z^{-1}) + K_p T_i 2 (1 - z^{-1})) \cdot T_s (1 + z^{-1})}{T_i 2 (1 - z^{-1}) \cdot T_s (1 + z^{-1})} \quad (54)$$

Expandiendo los términos y reduciendo factores comunes:

$$\frac{K_p T_s + 2K_p T_i + K_p T_s z^{-1} - 2K_p T_i z^{-1}}{2T_i (1 - z^{-1})} \quad (55)$$

Finalmente, factorizando los términos del numerador podemos obtener los valores de K_{N0} y K_{N1} , de forma que:

$$K_{N0} = K_p \left(\frac{T_s + 2T_i}{2T_i} \right) \quad (56)$$

$$K_{N1} = K_p \left(\frac{T_s - 2T_i}{2T_i} \right) \quad (57)$$

Por lo tanto, los valores de K_{N0} y K_{N1} calculados para el sistema mecánico y el sistema eléctrico se pueden observar en la tabla 3.2.0.1

	Lazo mecánico	Lazo eléctrico
K_{N0}	$7.60 \cdot 10^{-3}$	0.168
K_{N1}	$-7.50 \cdot 10^{-3}$	-0.156

Tabla 3.2.0.1: Valores de ganancias discretizadas

3.3. Controlador discreto con Anti Wind-Up

En algunos casos, el controlador intentará aplicar una actuación con un valor elevado que supera los límites de la planta, la parte integral acumulará el error durante todo el periodo que se mantenga saturada la salida del controlador, por esto, se debe implementar un sistema de anti-enrollamiento o “Anti Wind-Up” que desactiva la parte integral al saturar la actuación, de esta forma no se acumula el error.

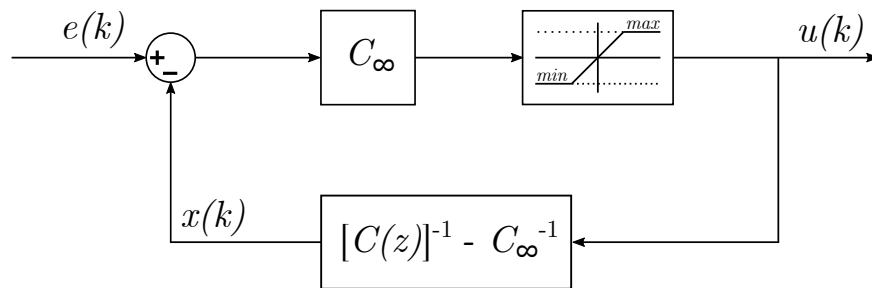


Fig. 3.3.0.1: Diagrama de bloques del controlador con Anti Wind-Up.

La actuación del controlador se define como:

$$u(k) = C_\infty (e(k) - x(k)) \quad (58)$$

Donde C_∞ es la ganancia de frecuencia infinita del controlador, y en un controlador PI equivale a la ganancia proporcional (K_p) y en un controlador discretizado equivale a K_{N0} . Por otro lado $C(z)^{-1}$ es:

$$C(z)^{-1} = \left[\frac{K_{N0} + K_{N1} \cdot z^{-1}}{1 - z^{-1}} \right]^{-1} \quad (59)$$

La función de realimentación entre $u(k)$ y $x(k)$ esta dada por:

$$\frac{x(k)}{u(k)} = \frac{-\frac{(K_{N0} + K_{N1})}{K_{N0}^2} \cdot Z^{-1}}{1 - \frac{K_{N1}}{K_{N0}} \cdot Z^{-1}} \quad (60)$$

Se puede re-parametrizar la ecuación anterior para simplificar la expresión asignando los valores:

$$PIAWU_{N1} = \frac{-(K_{N0} + K_{N1})}{K_{N0}^2} \quad (61)$$

$$PIAWU_{D1} = \frac{K_{N1}}{K_{N0}} \quad (62)$$

Por lo tanto:

$$x(k) (1 - PIAWU_{D1} \cdot Z^{-1}) = u(k) \cdot PIAWU_{N1} \cdot z^{-1} \quad (63)$$

Despejando $x(k)$:

$$x(k) = u(k-1) \cdot PIAWU_{N1} + x(k-1)PIAWU_{D1} \quad (64)$$

3.4. Resultados de simulación FOC

Se realizó una prueba de respuesta dinámica con diferentes valores de velocidad de referencia y torque, los resultados de esta prueba se pueden observar en la figura 3.4.1.1. Luego, en la figura 3.4.2.1 se observa el comportamiento en estado estacionario durante el periodo $t = 0.9s$ y $t = 1s$. Además, en la figura 3.4.3.1 se puede observar el comportamiento del control por campo orientado frente a diferentes valores de frecuencia de conmutación del PWM.

Para la prueba dinámica (Fig.3.4.2.1), se comenzó con una velocidad de referencia de 4200RPM, con un torque de carga igual al 30 % del nominal, luego, a los 500ms se aumenta el torque de carga al 80 % del nominal. En el tiempo igual a 1 segundo se hace una inversión de giro hasta los -2000RPM con un 60 % de torque. Finalmente, en el segundo 1.5 se vuelve a invertir el giro a 2000RPM, manteniendo el el torque de carga al 60 %.

Durante la fase inicial se observa un ligero sobrepaso en respuesta al aumento del ancho de banda que apresura el seguimiento de referencia, se logra un seguimiento estable y sin error en estado estacionario, luego, durante el aumento de carga se observa un aumento en la corriente de fase y nuevamente el se logra una velocidad igual a la de referencia sin error. Cuando se llega al periodo $t = 1s$, se observa el mismo efecto que en inicio de la simulación, un ligero sobrepaso de la velocidad en sentido contrario, además, un aumento instantáneo de las corrientes de fase y una caída en la corriente q y el torque de la máquina, producto de la inversión del sentido de aplicación del torque de carga.

Luego, se observa una saturación en la salida del controlador PI de velocidad durante la inversión de giro en el instante $t = 1.5$, debido al alto torque requerido para realizar la inversión mientras se mantiene el torque de carga, esto que genera una respuesta lineal en la velocidad con el "Anti Wind-Up" del controlador y una amplitud constante en las corrientes de fase.

En estado estacionario se observa un excelente seguimiento de referencia con un error menor al 0.4 %. Las corrientes de fase presentan un comportamiento balanceado y de amplitud constante, y se observa un ligero error promedio entre el torque de carga aplicado y el medido por la maquina controlada debido a la variación de la corriente q de la cual depende esta estimación .

3.4.1. Respuesta dinámica

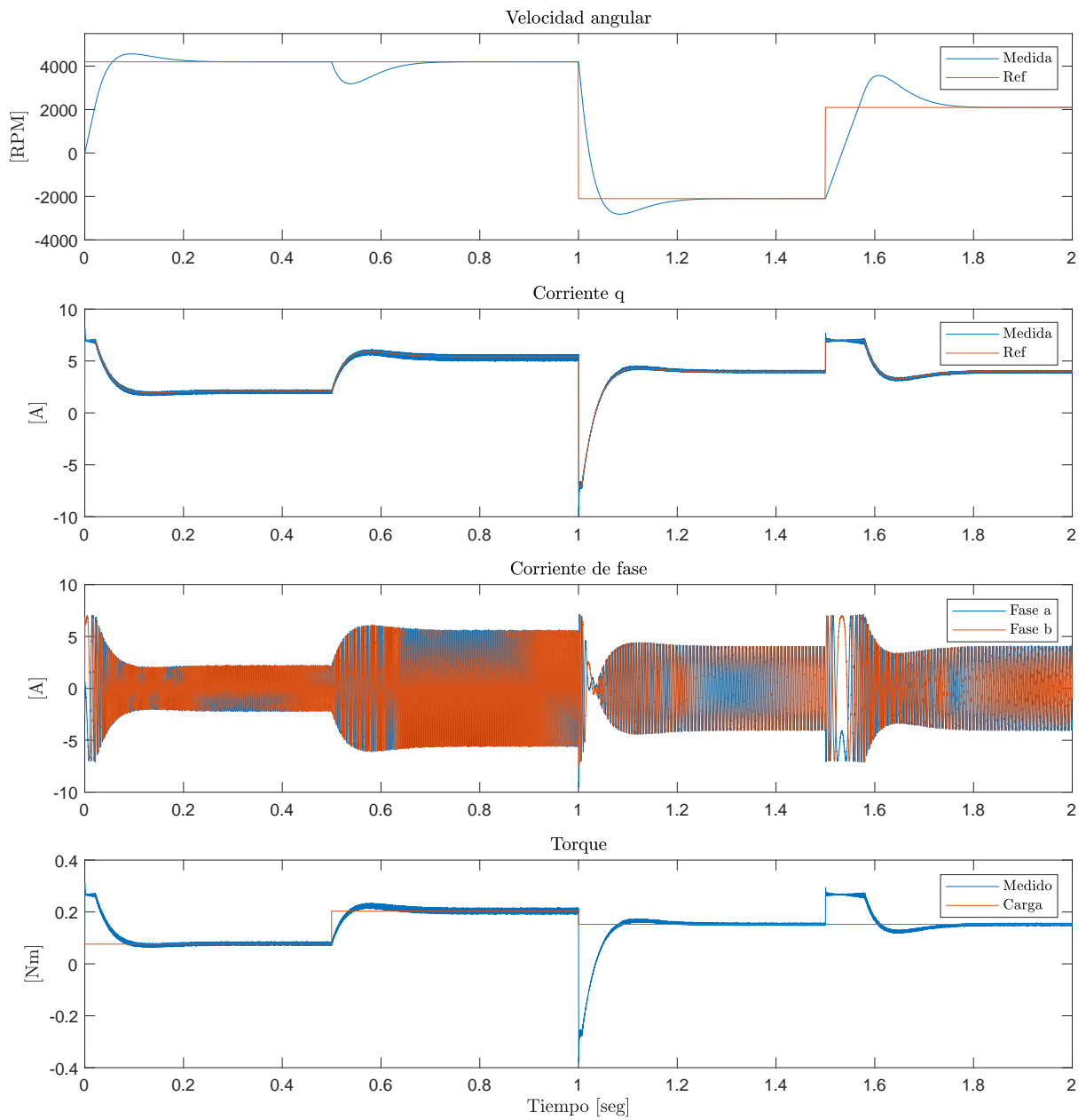


Fig. 3.4.1.1: Resultados de respuesta dinámica de control por campo orientado.

3.4.2. Estado estacionario

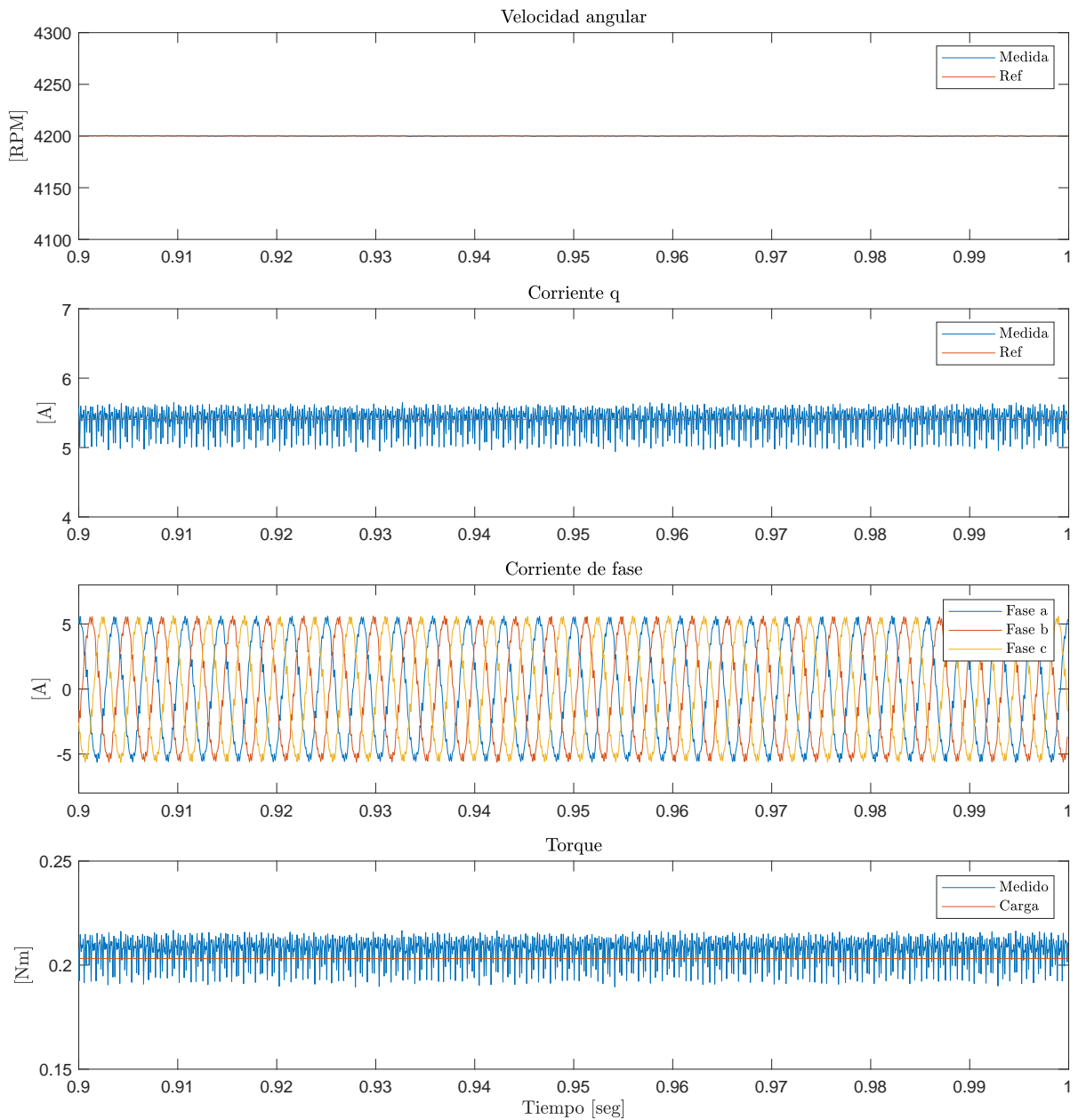


Fig. 3.4.2.1: Comportamiento en estado estacionario de control por campo orientado.

3.4.3. Distorsión armónica total

Para obtener esta curva, se realizaron 6 simulaciones de control por campo orientado utilizando diferentes frecuencias de la señal portadora del PWM, la selección de las frecuencias se basó en las frecuencias de conmutación que se obtuvieron de los métodos de control predictivos, debido a que en estos métodos la frecuencia de conmutación no se puede alterar directamente, sino que depende del tiempo de muestreo del controlador y el factor de peso W que se aplique en la función de costo del controlador. En la sección 5.4 se realizan comparaciones directas con los demás métodos propuestos.

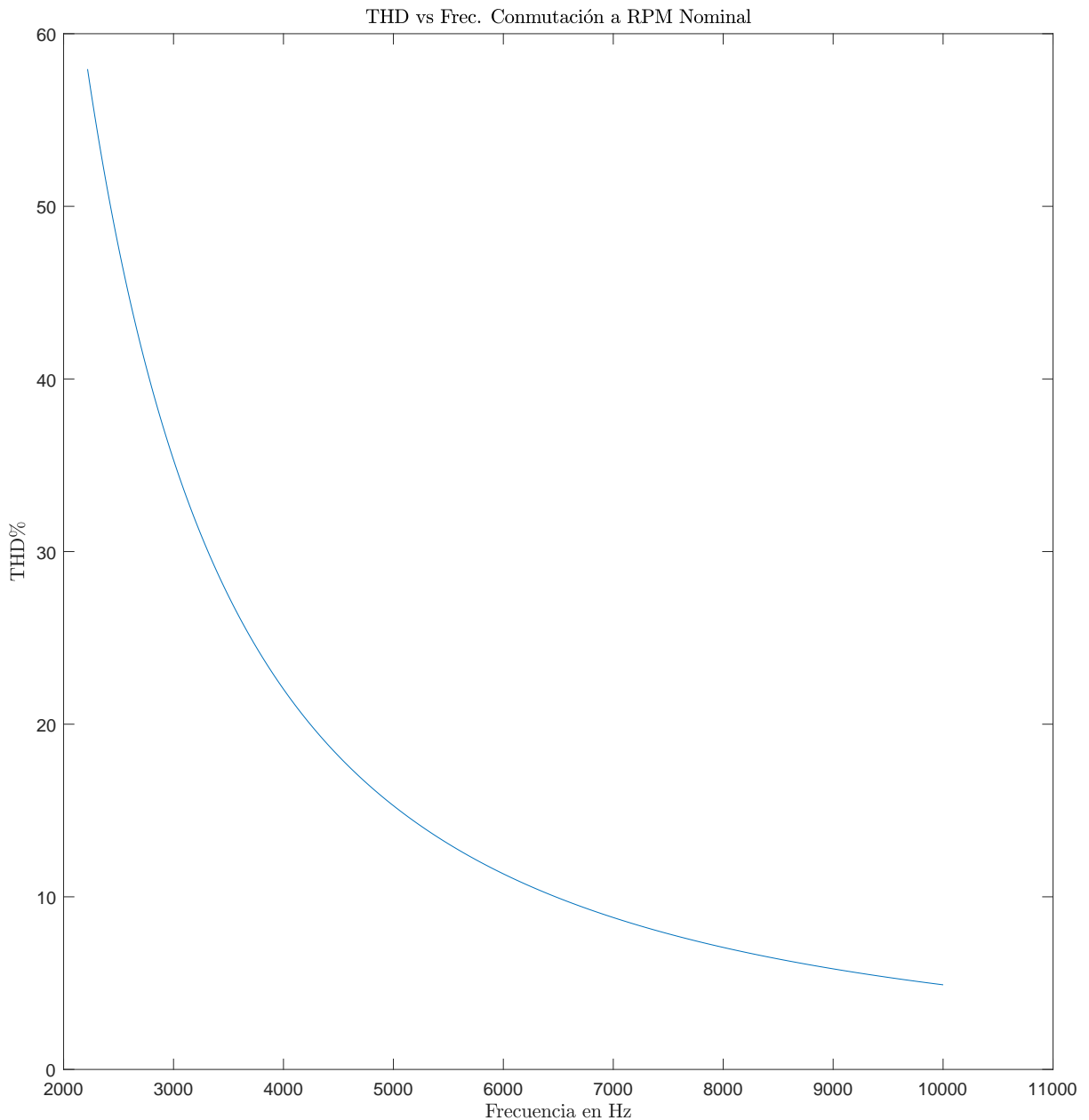


Fig. 3.4.3.1: Resultados de respuesta dinámica de control por campo orientado.

4. Simulación control predictivo de estados finitos

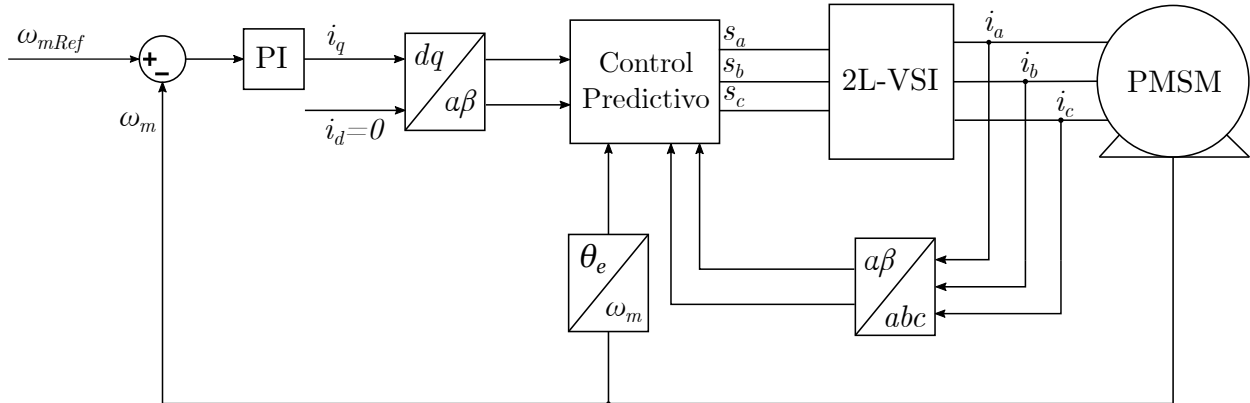


Fig. 4.0.0.1: Diagrama de control predictivo.

El método de control predictivo se enfoca en resolver un problema de optimización, donde la función a optimizar se denomina “función de costo”, el valor óptimo de esta función determina la actuación que aplicará el controlador sobre la planta. Los valores que se comparan en el problema de optimización se calculan en cada tiempo de muestreo, basándose en el modelo de la PMSM, donde se predicen los estados futuros del sistema, en relación al estado actual medido.

En el caso de un controlador predictivo aplicado a un inversor de voltaje, la actuación del controlador esta acotada a un conjunto finito de estados de conmutación, particularmente en un inversor de dos niveles, se tienen 8 estados posibles que generan diferentes señales de voltaje, estos estados pueden representarse como:

$$S = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (65)$$

Donde cada fila representa una pierna del inversor y cada columna representa una posible combinación de estados de los semiconductores que conforman la parte superior en cada pierna del inversor, y el estado del inversor en la parte inferior es simplemente el valor negado del superior. La primera columna de la matriz hace que los semiconductores superiores estén apagados, y por consecuencia los de la parte inferior conduciendo, el efecto contrario se produce en la octava columna. Estas 2 columnas producen un voltaje nulo sobre la máquina eléctrica.

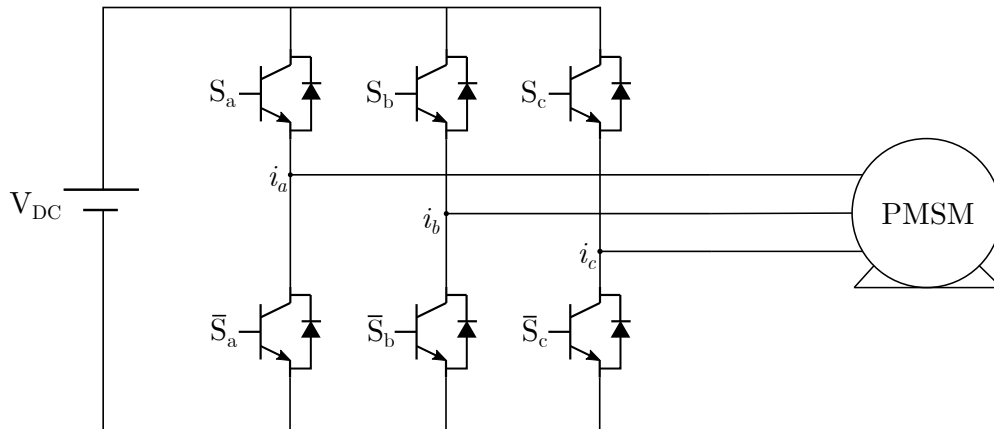


Fig. 4.0.0.2: topología de un convertidor trifásico de dos niveles.

4.1. Modelo predictivo

El modelo de la máquina desarrollado en la sección 2 representa la dinámica de la máquina síncrona de imanes permanentes. Estas ecuaciones se pueden escribir en forma de variables de estado. Esta representación permite simplificar las operaciones y además, permite obtener una discretización de forma sencilla. El modelo en variables de estado de la máquina síncrona de imanes permanentes representado en el plano $\alpha - \beta$ esta dado por:

$$\frac{d}{dt} \begin{bmatrix} i_{\alpha}(t) \\ i_{\beta}(t) \end{bmatrix} = A \begin{bmatrix} i_{\alpha}(t) \\ i_{\beta}(t) \end{bmatrix} + B \begin{bmatrix} V_{\alpha}(t) \\ V_{\beta}(t) \end{bmatrix} + F\omega_e \begin{bmatrix} \sin(\theta_e) \\ \cos(\theta_e) \end{bmatrix} \quad (66)$$

Donde ω_e es la velocidad angular eléctrica, y θ_e es el ángulo eléctrico. Las matrices A, B y F son:

$$A = \begin{bmatrix} -\frac{R}{L} & 0 \\ 0 & -\frac{R}{L} \end{bmatrix} \quad (67)$$

$$B = \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & \frac{1}{L} \end{bmatrix} \quad (68)$$

$$F = \begin{bmatrix} \frac{\lambda_{PM}}{L} & 0 \\ 0 & \frac{-\lambda_{PM}}{L} \end{bmatrix} \quad (69)$$

4.2. Discretización del modelo

Típicamente, la forma mas común de discretizar un modelo, es usar el método de Euler, debido a su simplicidad y rapidez en la aproximación al tiempo discreto. Este método de basa en en la expansión de series de Taylor de primer orden.

Para un tiempo de muestreo pequeño, la aproximación de serie de Taylor es valida, por lo tanto las ecuaciones quedan:

$$\frac{i_{\alpha}(t + \Delta t) - i_{\alpha}(t)}{\Delta t} = \frac{V_{\alpha}(t)}{L} - \frac{R}{L}i_{\alpha}(t) + \frac{\lambda_{PM}}{L}\omega_e(t)\sin(\theta) \quad (70)$$

$$\frac{i_\beta(t + \Delta t) - i_\beta(t)}{\Delta t} = \frac{V_\beta(t)}{L} - \frac{R}{L}i_\beta(t) - \frac{\lambda_{PM}}{L}\omega_e(t)\cos(\theta) \quad (71)$$

Si Δt es un periodo de muestreo fijo igual a T_s y es lo suficientemente pequeño como para despreciar el cambio de velocidad angular ω_e , entonces la ecuación discretizada para un periodo $k + 1$ es:

$$i_\alpha(k + 1) = \frac{T_s V_\alpha(k)}{L} + i_\alpha(k) - \frac{T_s R}{L}i_\alpha(k) + T_s \frac{\lambda_{PM}}{L}\omega_e \sin(\theta) \quad (72)$$

$$i_\beta(k + 1) = \frac{T_s V_\beta(k)}{L} + i_\beta(k) - \frac{T_s R}{L}i_\beta(k) - T_s \frac{\lambda_{PM}}{L}\omega_e \cos(\theta) \quad (73)$$

Reordenando las ecuaciones:

$$i_\alpha(k + 1) = \left(1 - \frac{T_s R}{L}\right) i_\alpha(k) + \frac{T_s}{L}V_\alpha(k) + T_s \frac{\lambda_{PM}}{L}\omega_e \sin(\theta) \quad (74)$$

$$i_\beta(k + 1) = \left(1 - \frac{T_s R}{L}\right) i_\beta(k) + \frac{T_s}{L}V_\beta(k) - T_s \frac{\lambda_{PM}}{L}\omega_e \cos(\theta) \quad (75)$$

La representación en variables de estado del modelo discretizado queda como:

$$\begin{bmatrix} i_\alpha(k + 1) \\ i_\beta(k + 1) \end{bmatrix} = (I + T_s A) \begin{bmatrix} i_\alpha(k) \\ i_\beta(k) \end{bmatrix} + T_s B \begin{bmatrix} V_\alpha(k) \\ V_\beta(k) \end{bmatrix} - F \omega_e \begin{bmatrix} \sin(\theta_e) \\ \cos(\theta_e) \end{bmatrix} \quad (76)$$

Donde I representa la matriz identidad de 2×2 .

4.3. Función de costo

El esquema de control predictivo necesita una función que optimice los valores calculados y aplique la señal de actuación adecuada que se ajuste mejor a la referencia. La función de costo se puede representar como:

$$g = (i_{\alpha Ref} - i_\alpha(k + 2))^2 + (i_{\beta Ref} - i_\beta(k + 2))^2 + W \|S(abc, n_k) - S(abc, n_{k-1})\| \quad (77)$$

La función de costo g se calcula cada iteración con los valores predichos para un tiempo de muestreo $k + 1$. El primer término calcula el error entre la corriente de referencia α y el valor de la corriente predicha, el segundo termino calcula el error entre la corriente de referencia β y la predicha, y el tercer termino calcula la cantidad de conmutaciones en cada pierna comparando el estado de conmutación actual (k) con el aplicado en el periodo anterior ($k - 1$), este termino intenta reducir la cantidad de conmutaciones necesarias para mantener el seguimiento de referencia y es multiplicado por un factor de peso W para ajustar su participación en el proceso de optimización .

4.4. Resultados de simulación MPC

Se realizó la misma prueba de respuesta dinámica que para el método anterior, los resultados de esta prueba se pueden observar en la figura 4.4.1.1.

El comportamiento es igual que el caso del control por campo orientado, ya que el controlador de velocidad es el mismo para todos los métodos, el cambio se produce en el lazo eléctrico, donde ahora se controla con las corrientes de referencia α y β , que presentan buen seguimiento de referencia en estado estacionario (Fig.4.4.2.1).

Se observa un comportamiento similar que el caso anterior de FOC, un ligero sobrepaso para luego estabilizarse sin error en estado estacionario, en este caso las corrientes controladas son α y β en vez de $d-q$. Como el lazo de control de velocidad es el mismo, no se aprecia un cambio sustancial en la dinámica de la máquina. Si existe alguna diferencia, esta será mas evidente en los análisis de distorsión armónica.

En estado estacionario se observa un ligero ruido en en la velocidad medida durante algunos milisegundos, pero es despreciable al ser un error menor al 1% de la velocidad de referencia y en promedio es casi nula.

4.4.1. Respuesta dinámica

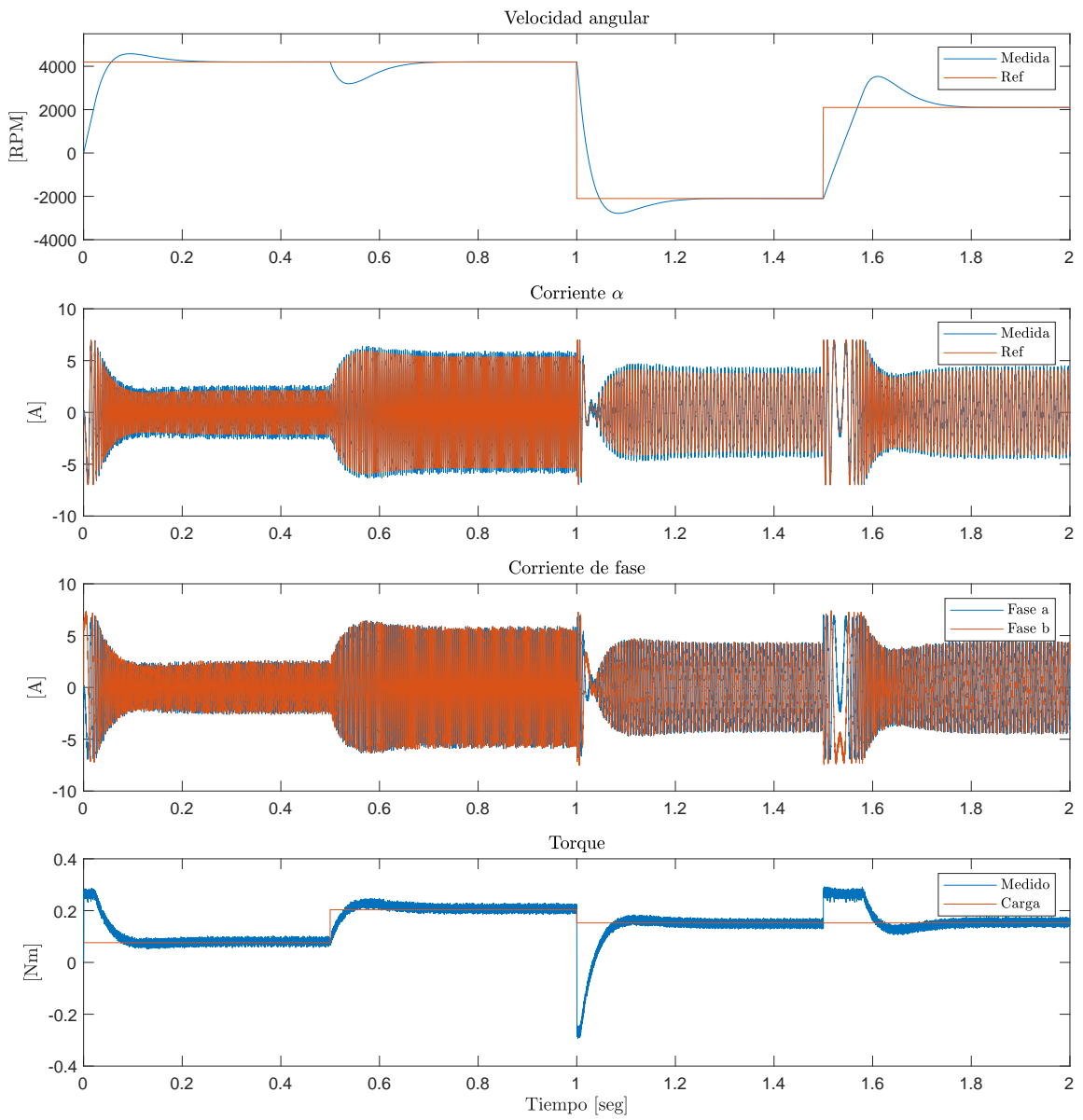


Fig. 4.4.1.1: Resultados de respuesta dinámica de control predictivo de estados finitos.

4.4.2. Estado estacionario

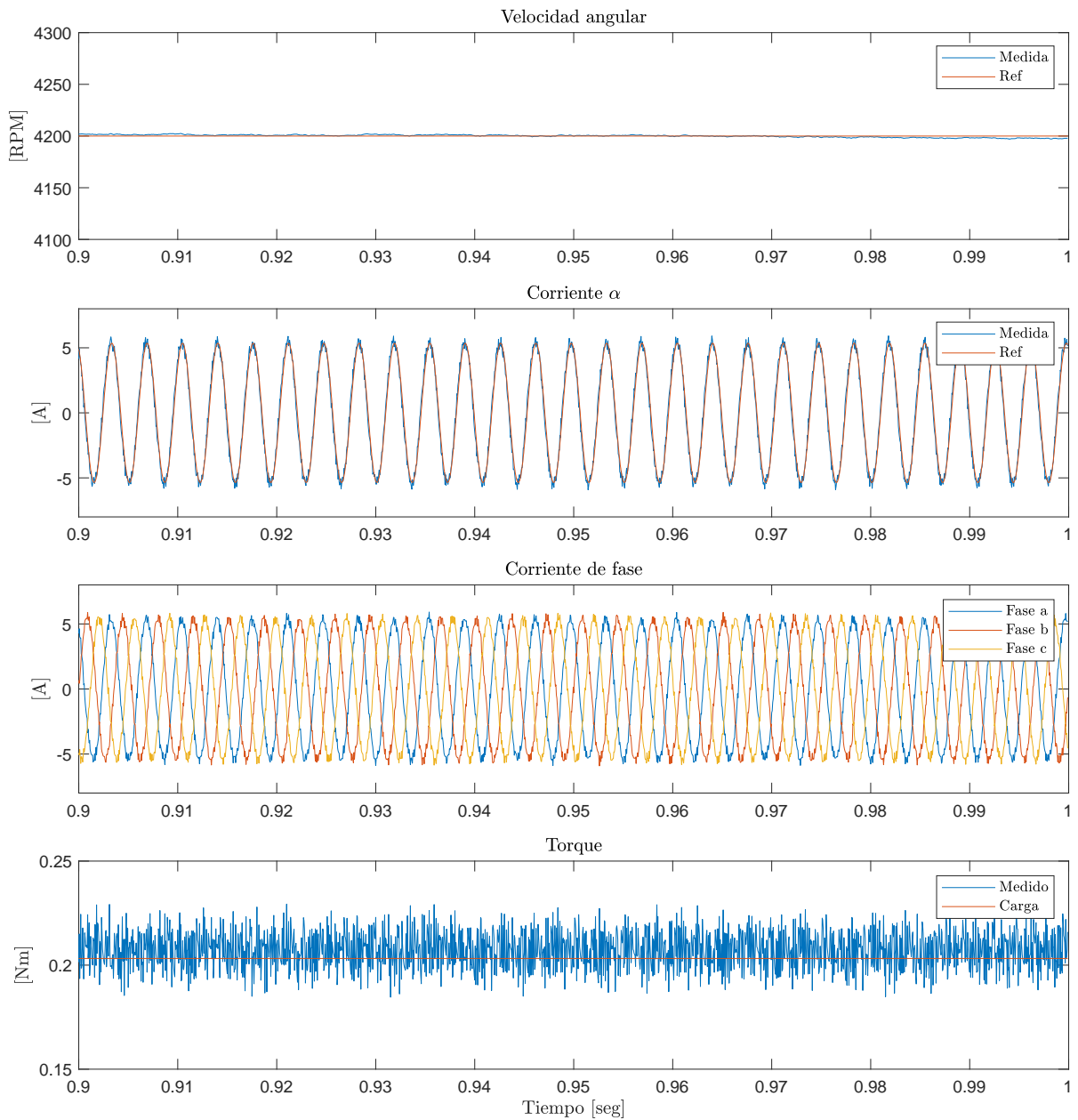


Fig. 4.4.2.1: Comportamiento durante estado estacionario de control predictivo de estados finitos.

4.4.3. Distorsión armónica total versus frecuencia de conmutación promedio

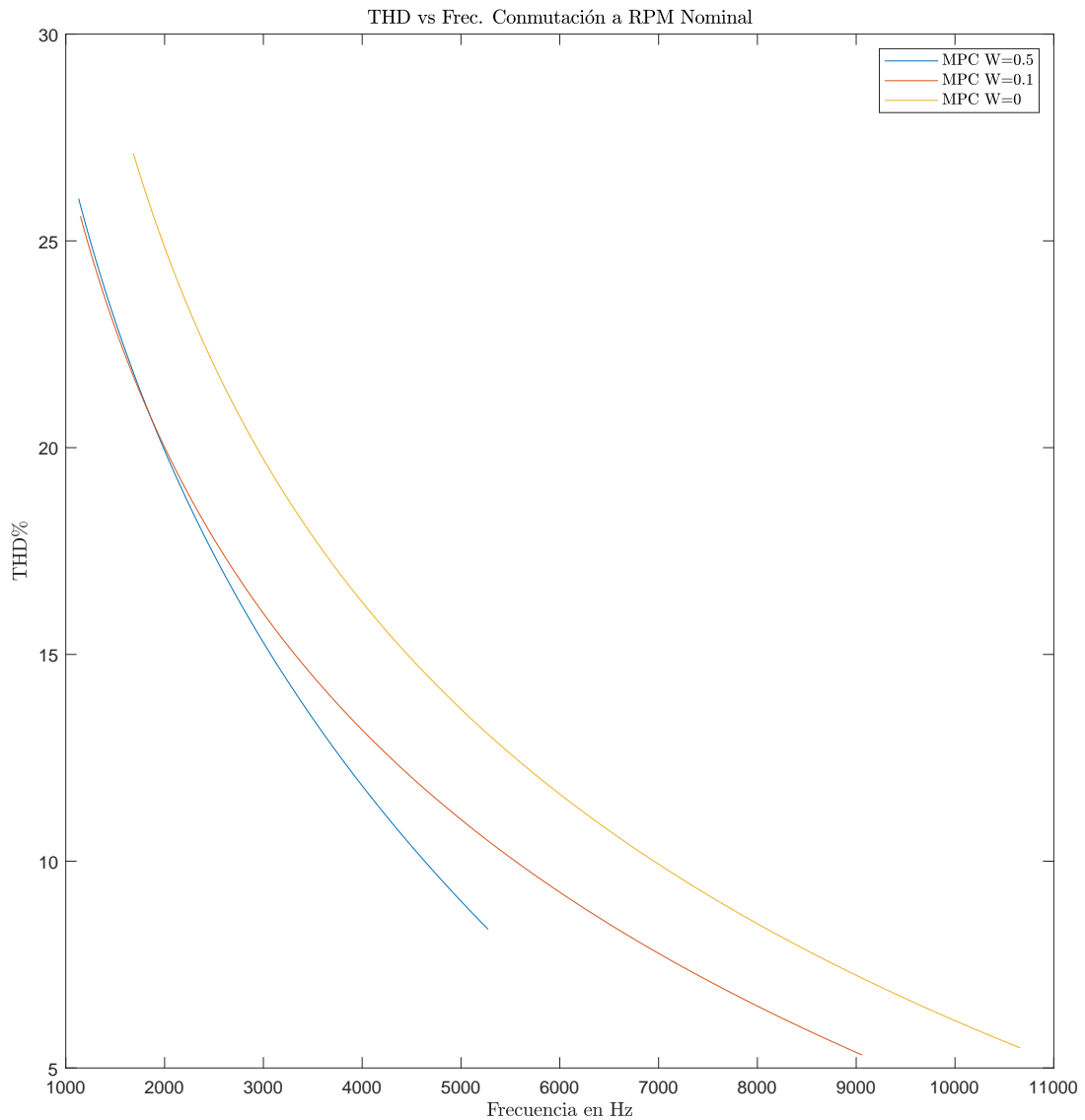


Fig. 4.4.3.1: Comparación de frecuencia de conmutación ante distintos valores de peso W en control predictivo de estados finitos.

Para realizar esta comparación se tomó 6 muestras de cada controlador a distintos tiempos de muestreo en un rango de $20\mu s$ hasta $120\mu s$ con un valor de peso W diferente para un total de 18 simulaciones, luego, los datos fueron filtrados e interpolados para generar una curva ajustada que representa el comportamiento del controlador comparando la frecuencia de conmutación con la distorsión armónica total.

Se puede observar que ante un mismo tiempo de muestreo, el controlador con un mayor factor de peso produce menor número de conmutaciones en el inversor de voltaje, sin embargo genera una distorsión armónica mayor.

5. Simulación control predictivo de horizonte lejano

El método de control predictivo de horizonte lejano funciona bajo el mismo principio que el control predictivo clásico, la mayor diferencia es la cantidad de predicciones en el tiempo que se calculan en un tiempo de muestreo para obtener los valores a optimizar. Este método no ha sido ampliamente utilizado debido a su alto costo computacional. Sin embargo, gracias al acelerado desarrollo tecnológico, los dispositivos digitales programables poseen una capacidad de procesamiento cada vez mayor, lo que permite ejecutar este tipo de algoritmos de control de forma “On-Line”.

El modelo que se utiliza para el cálculo de predicciones es el mismo que se usa para el control predictivo desarrollado en la sección anterior, dado por:

$$\begin{bmatrix} i_\alpha(k+1) \\ i_\beta(k+1) \end{bmatrix} = (I + T_s A) \begin{bmatrix} i_\alpha(k) \\ i_\beta(k) \end{bmatrix} + T_s B \begin{bmatrix} V_\alpha(k) \\ V_\beta(k) \end{bmatrix} - F\omega_e \begin{bmatrix} \sin(\theta_e) \\ \cos(\theta_e) \end{bmatrix} \quad (78)$$

Sin embargo, se debe tener una consideración especial con los valores que son variantes en el tiempo cuando se consideran varios horizontes de predicción, como es el caso de este modelo en el plano $\alpha - \beta$. Particularmente el ángulo eléctrico de la máquina puede cambiar lo suficiente durante los periodos de predicción, como para afectar los valores de corriente calculados, es por esto, que se debe realizar una extrapolación de su valor para diferentes horizontes de predicción, simulando su valor en los periodos futuros.

5.1. Extrapolación de señales

Para señales sinusoidales, el uso de extrapolación puede compensar el retardo producido al comparar los valores medidos en un tiempo de muestreo (k) en relación a los valores predichos den un tiempo ($k + N$), donde N es la cantidad de horizontes de perdición, en particular, en esta simulación se propone un horizonte de $K + 4$, por lo tanto, se estaría comparando valores predichos en 4 periodos de muestreo futuros con referencias en el periodo actual, lo que podría generar un gran error en el seguimiento de la referencia.

Una posible solución para obtener el valor futuro de una señal sinusoidal utilizando el valor actual, es usar la extrapolación de Lagrange, expresada como:

$$x(k+1) = \sum_{l=0}^n (-1)^{n-l} \binom{n+1}{l} x(k+l-n) \quad (79)$$

Para señales sinusoidales es recomendable el uso de $n=2$ [18], por tanto la ecuación anterior se puede representar como:

$$x(k+1) = 3x(k) - 3x(k-1) + x(k-2) \quad (80)$$

Si se requiere extrapolar para tiempos futuros como ($k+2$), se puede hacer un paso en el tiempo de un periodo de muestro:

$$x(k+2) = 3x(k+1) - 3x(k) + x(k-1) \quad (81)$$

Y al reemplazar la ecuación (80) en (81) se puede calcular el valor de la señal en $k+2$ usando solo valores presentes y pasados de la señal.

$$x(k+2) = 6x(k) - 8x(k-1) + 3x(k-2) \quad (82)$$

Utilizando el mismo procedimiento, se puede extrapolar la señal para un tiempo futuro arbitrario, en particular, para un tiempo $k + 3$ y $k + 4$, las ecuaciones se pueden representar como:

$$x(k + 3) = 10x(k) - 15x(k - 1) + 6x(k - 2) \quad (83)$$

$$x(k + 4) = 15x(k) - 24x(k - 1) + 10x(k - 2) \quad (84)$$

5.2. Función de costo

La función de costo para el método de control predictivo de horizonte lejano debe lidiar con los valores predichos durante varios periodos de tiempo, por esto la nueva función de costo ahora es una suma de todos los costos de cada horizonte de predicción, esta nueva función se puede representar como:

$$g_{LH} = \sum_{k=0}^N (i_{\alpha Ref}(k + 1) - i_{\alpha}(k + 1))^2 + (i_{\beta Ref}(k + 1) - i_{\beta}(k + 1))^2 + W \|S(abc, n_k) - S(abc, n_{k-1})\| \quad (85)$$

La función de costo g_{LH} se calcula cada tiempo de muestreo como la suma de los costos de las iteraciones anteriores desde el periodo inicial (k) hasta el horizonte de predicción deseado ($k + N$). Esto produce que el calculo crezca de forma potencial, con cada horizonte de predicción. Por ejemplo, se dijo anteriormente que un inversor de dos niveles posee ocho estados posibles de conmutación, por lo tanto, con un solo horizonte de predicción y considerando la compensación de retardo que desplaza la predicción por un periodo de muestreo ($k + 2$) se tiene una función de costo que calcula 8 valores en un periodo de muestreo. Sin embargo si se agrega otro horizonte de predicción ($k + 3$), ahora se tienen 64 valores. Luego, con cada horizonte subsecuente se multiplica por 8 el valor de predicciones que deben ser optimizadas.

5.3. Resultados de simulación LH-MPC

Se realizó una prueba de respuesta dinámica con diferentes valores de velocidad de referencia y torque, los resultados de esta prueba se pueden observar en la figura 5.3.1.1. En la figura 5.3.2.1 se observa el comportamiento en estado estacionario durante el periodo $t = 0.9s$ y $t = 1s$. Además, en la figura 5.3.3.1 se realiza una comparación entre los valores de distorsión armónica del control predictivo frente a diferentes valores de peso W en la función de costo.

Al igual que en el caso de FS-MPC, en esta comparación se tomo 6 muestras de cada controlador a distintos tiempos de muestreo en un rango de $20\mu s$ hasta $120\mu s$ con un valor de peso W diferente para un total de 18 simulaciones, los datos fueron filtrados e interpolados para generar una curva ajustada que representa el comportamiento del controlador comparando la frecuencia de conmutación con la distracción armónica total.

Se puede observar el mismo efecto en las curvas del controlador, donde a mayor factor de peso, la curva de desplaza hacia la izquierda, indicando que se producen menores conmutaciones a cambio de una mayor distorsión armónica.

5.3.1. Respuesta dinámica

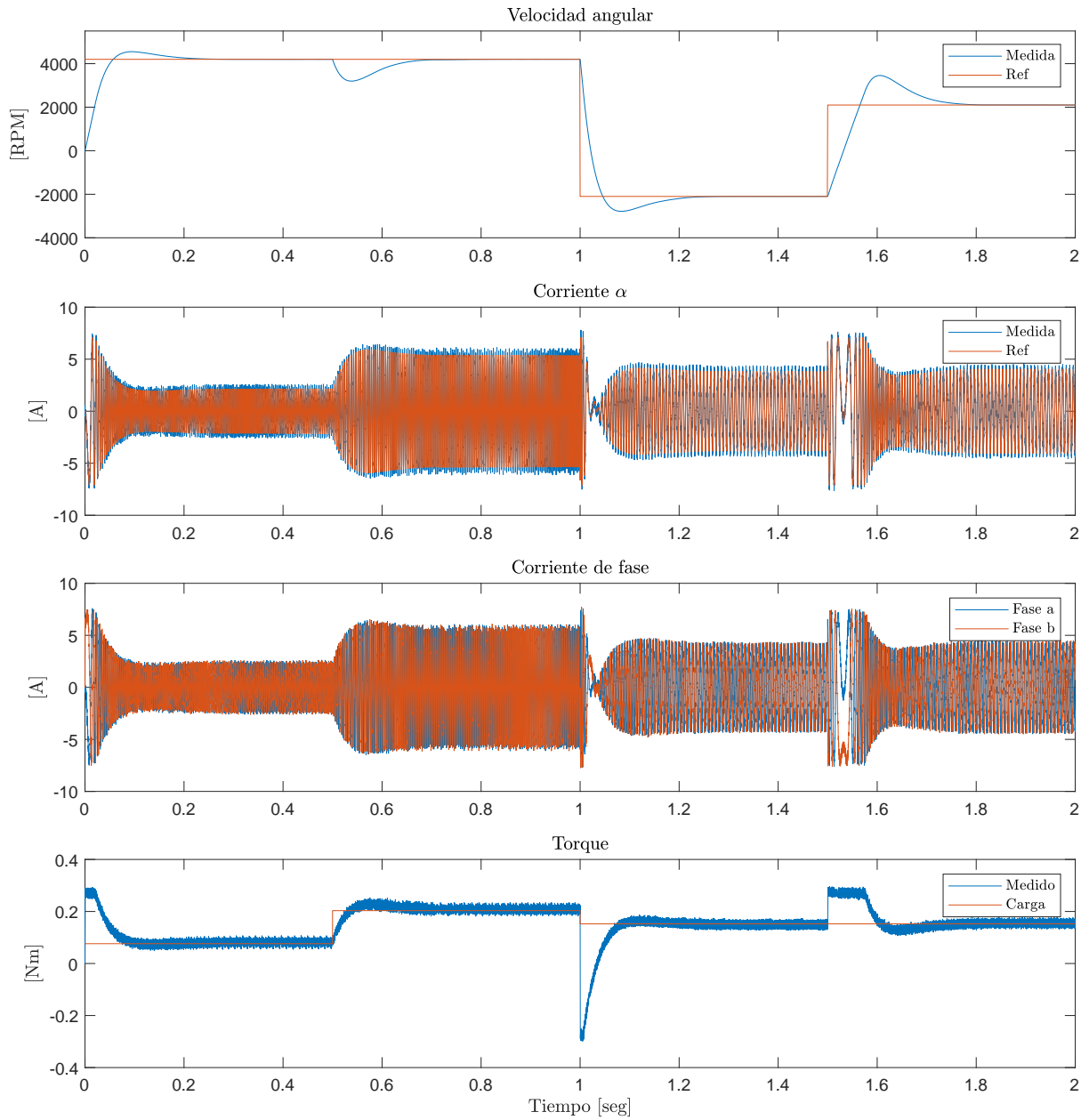


Fig. 5.3.1.1: Resultados de respuesta dinámica de control predictivo de horizonte lejano.

5.3.2. Estado estacionario

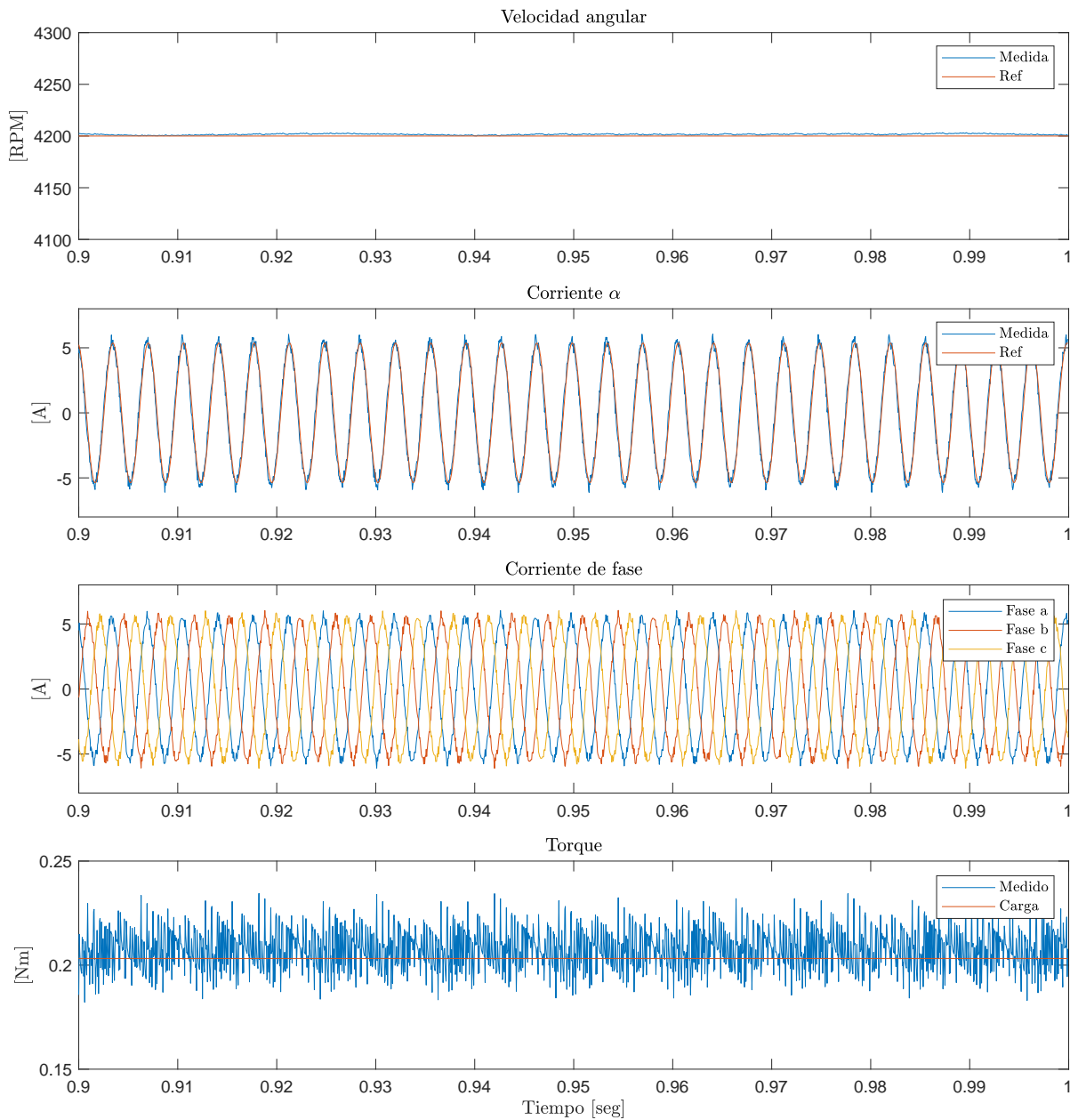


Fig. 5.3.2.1: Comportamiento durante estado estacionario de control predictivo de horizonte lejano.

5.3.3. Distorsión armónica total

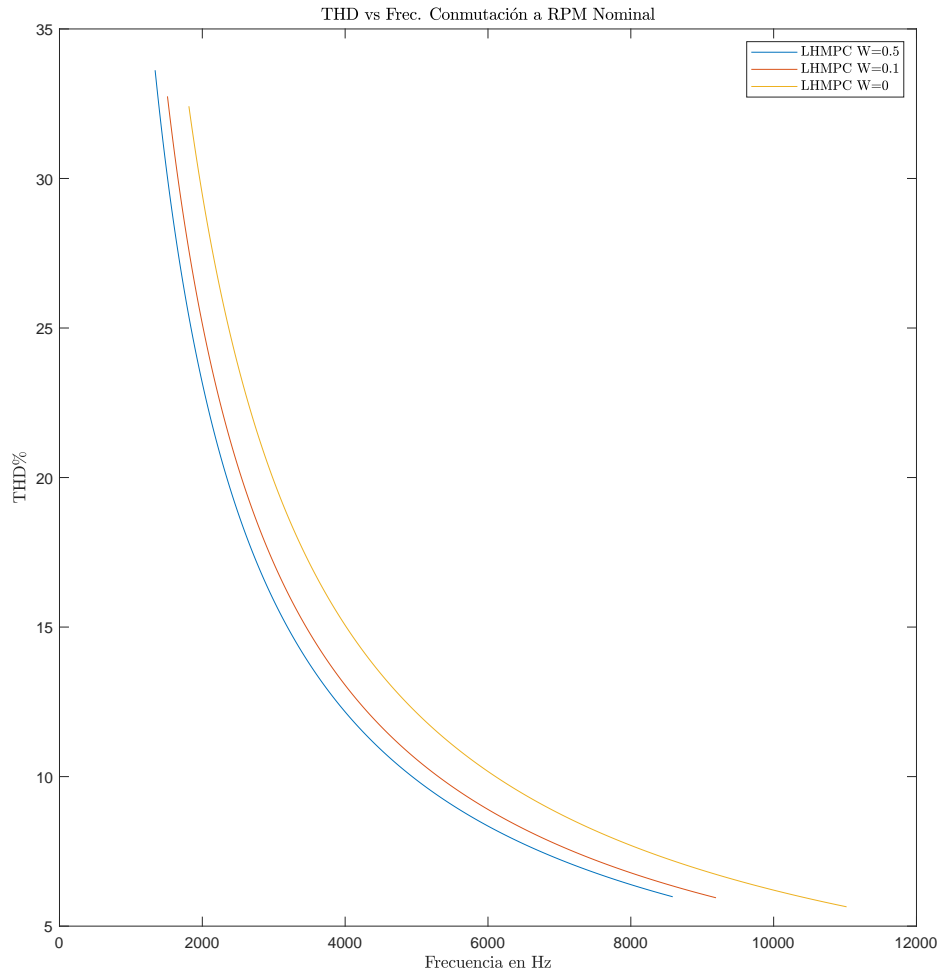


Fig. 5.3.3.1: Comparación de frecuencia de conmutación ante distintos valores de peso W en control predictivo de horizonte lejano.

5.4. Comparación de los tres métodos

En la figura 5.4.0.1 se observan las comparaciones del método de control FOC, FS-MPC y LH-MPC con distintos valores del factor de peso W , en primera instancia se aprecia que ante un factor de peso igual a cero los tres métodos presentan una curva similar en el rango de 6000 a 11000Hz, no se puede decir objetivamente que un método sea mejor que el otro, debido a la exactitud de los datos obtenidos, y que todos presentan un mejor comportamiento que los demás en algún tramo de las frecuencias de conmutación.

Lo que sí es claro, es que a bajas frecuencias de conmutación el control por campo orientado presenta una distorsión armónica mucho más alta que los métodos predictivos, lo que hace fácil la selección de estos métodos en estas condiciones.

Por otra parte, frente a un factor de peso distinto de cero, el LH-MPC es muy similar al FS-MPC y su utilización no demuestra una mejora substancial ante los demás métodos para el caso estudiado, por lo que su elección deberá ser estudiada dependiendo del caso específico de máquina eléctrica y el controlador físico que se desee utilizar.

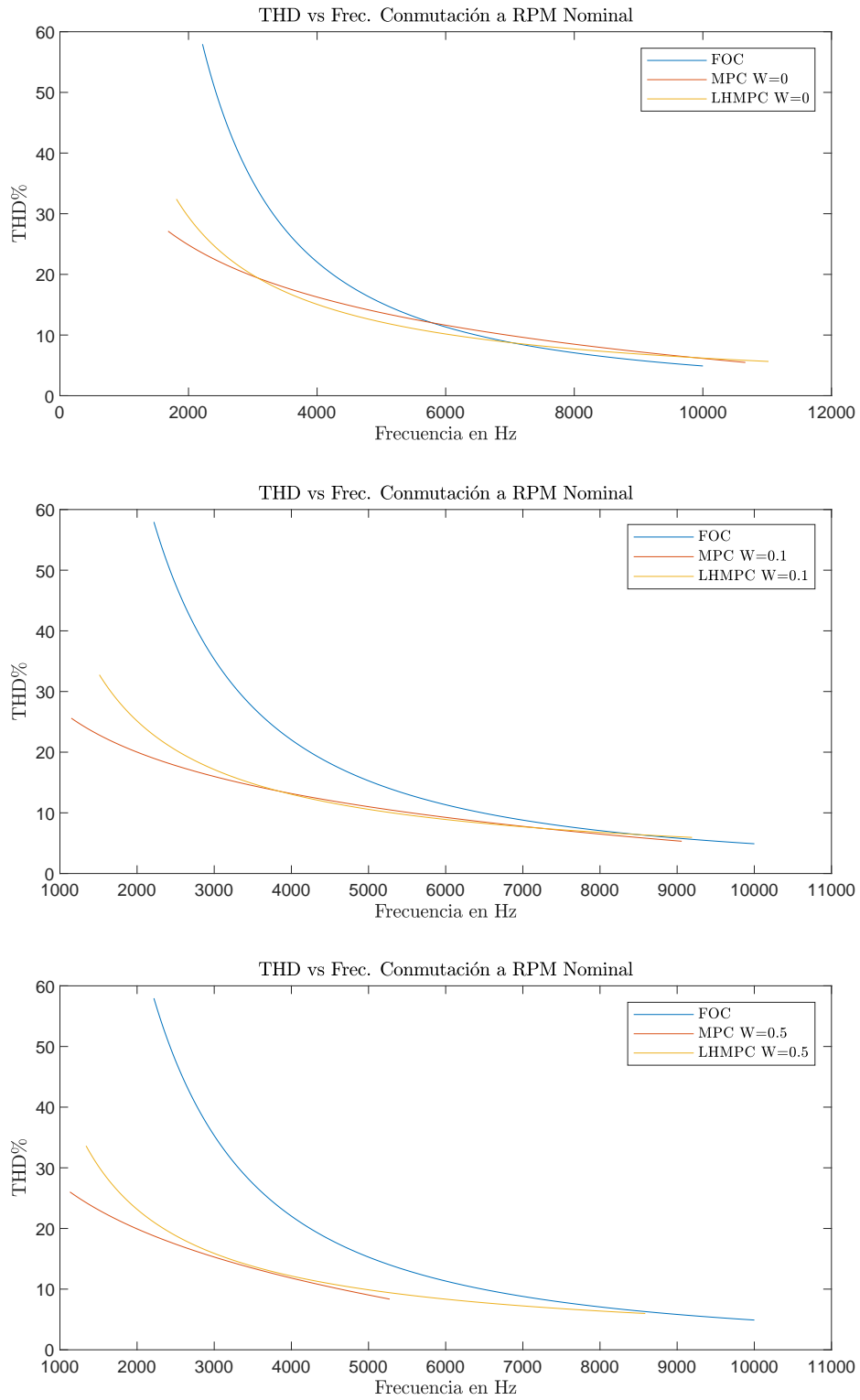


Fig. 5.4.0.1: Comparación de THD entre los tres métodos de control propuestos.

6. Validación experimental

6.1. Implementación del banco de pruebas

Para la implementación del banco de pruebas se utilizaron componentes comerciales diseñados para trabajar juntos, estos componentes son de la marca Texas Instruments y consisten en un Launchpad modelo LAUNCHXL-F28379D (Fig. 6.1.0.1), que posee 2 núcleos y una frecuencia de 200MHz.

Los inversores son el modelo BOOSTXL-3PHGanInv (Fig. 6.1.0.2) que es un inversor de dos niveles con un límite de corriente de 16A, y filtro de sensores. Los motores son de un kit de desarrollo creado por Texas Instruments, pero el fabricante es Teknic y el modelo es M-3210P-LN-04K (Fig. 6.1.0.3).

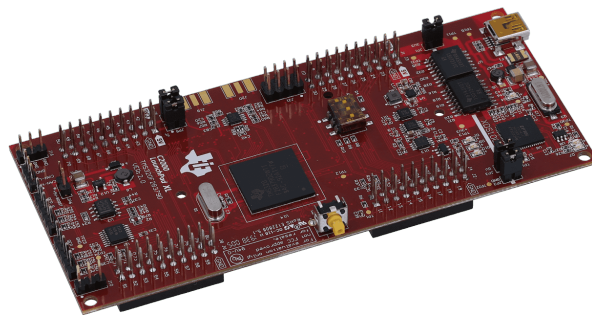


Fig. 6.1.0.1: Tarjeta de desarrollo de la marca Texas Instruments modelo LAUNCHXL-F28379D

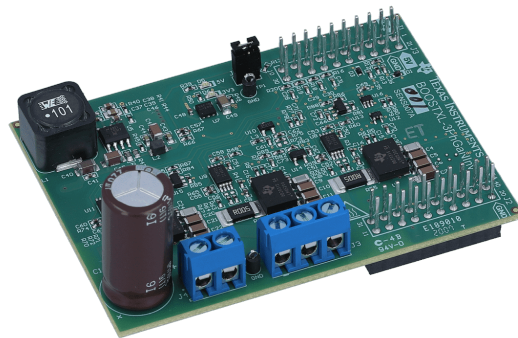


Fig. 6.1.0.2: Inversor de dos niveles de la marca Texas Instruments modelo BOOSTXL-3PHGanInv

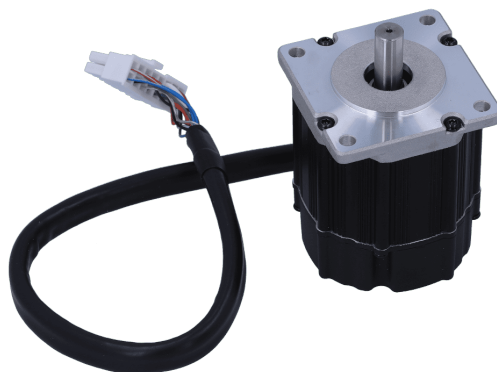


Fig. 6.1.0.3: PMSM de la marca Teknic modelo M-3210P-LN-04K

El banco de pruebas se implementó en el laboratorio de transición de energías (ETL), en la facultad de ingeniería de la Universidad de Talca. La figura 6.1.0.4 muestra como se dispusieron los elementos en el laboratorio para llevar a cabo las pruebas experimentales.

En la zona amarilla se encuentra el par de motores acoplados en un bloque de metal con una apertura, esto permite medir la velocidad física de los motores con un adhesivo reflectante pegado en el acople y un tacómetro laser.

En la zona celeste, se encuentran 2 inversores conectados a cada motor, además, se observan 2 sensores de corriente acoplados a una fase de cada inversor, estos sensores están conectados a un osciloscopio digital (Fig. 7.0.0.1) que permite observar las corrientes del motor controlado y el motor de carga, además, permite guardar las señales mostradas en una tabla de datos, para poder analizarlos posteriormente.

En la zona roja se encuentra el computador usado de host para controlar el launchpad, en la pantalla del computador se observa los datos que obtienen en tiempo real del funcionamiento del banco de pruebas, como velocidad de giro, corrientes de fase y torque eléctrico.

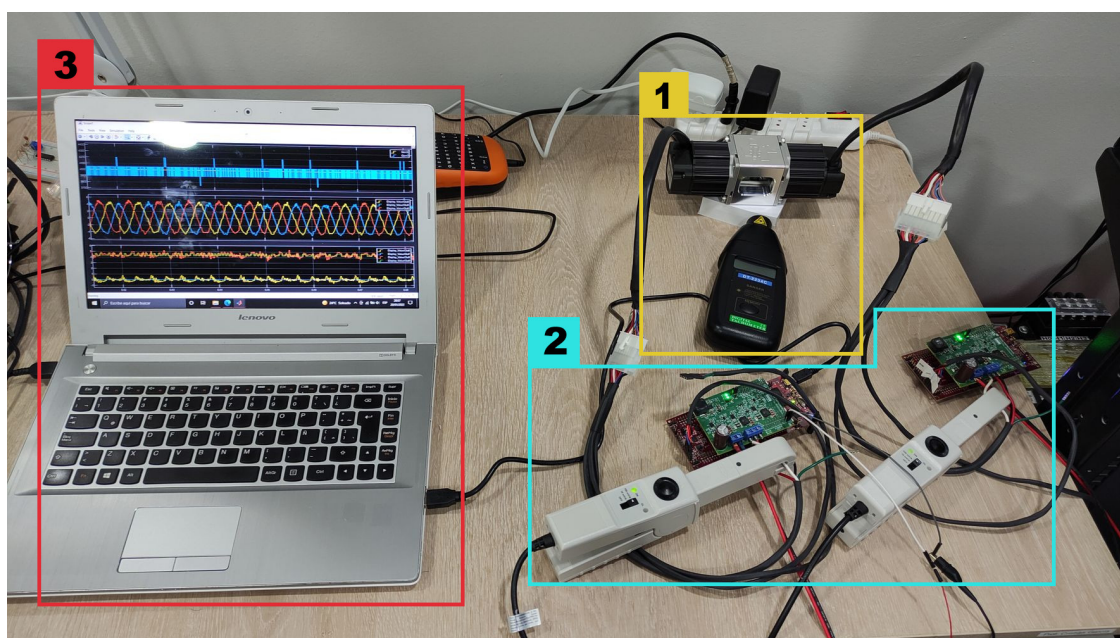


Fig. 6.1.0.4: Setup experimental. Zona 1 (en amarillo): Motores acoplados y tacómetro laser. Zona 2 (en celeste): par de inversores con sensores de corriente. Zona 3 (en rojo): Computador host recibiendo datos del launchpad.

6.2. Configuración del protocolo de comunicación

Para lograr recibir y enviar datos desde el microcontrolador y visualizarlos en host, se debe configurar la comunicación serial, este paso es crucial para que el controlador funcione de forma adecuada.

Primero se debe establecer la tasa de transferencia de datos en baudios, dependiendo de este valor, el microcontrolador adaptará la velocidad de procesamiento, con el fin de lograr una transferencia de datos correcta. Debido a este ajuste, si se elige una tasa de transferencia menor a la máxima que para el microcontrolador usado (Launchpad F28397D) es de 12×10^6 bits por segundo, el microcontrolador no operará al periodo de muestreo establecido en el código del programa. La figura 6.2.0.1 muestra donde se debe establecer la tasa de transferencia en la configuración de hardware en Simulink.

Para entablar la conexión, se debe utilizar el toolbox llamado “Motor Control Toolbox”, en el ambiente

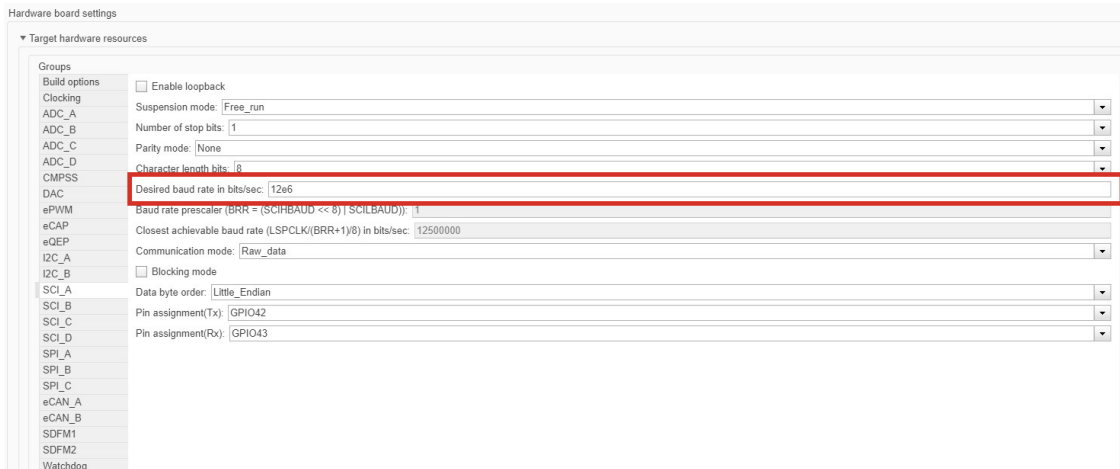


Fig. 6.2.0.1: Ventana de configuración de hardware, en rojo, la sección de tasa de transferencia de datos.

de Simulink, este permite utilizar bloques de host serial para la transmisión de datos entre los dispositivos. Se debe verificar que el puerto COM seleccionado en el bloque de host sea el correspondiente con el asignado por sistema operativo al conectar el launchpad por primera vez al computador.

La figura 6.2.0.2 muestra una interfaz de control, que permite cambiar la velocidad de giro del motor en tiempo real, y recibir datos desde el launchpad a través de la conexión USB serial.

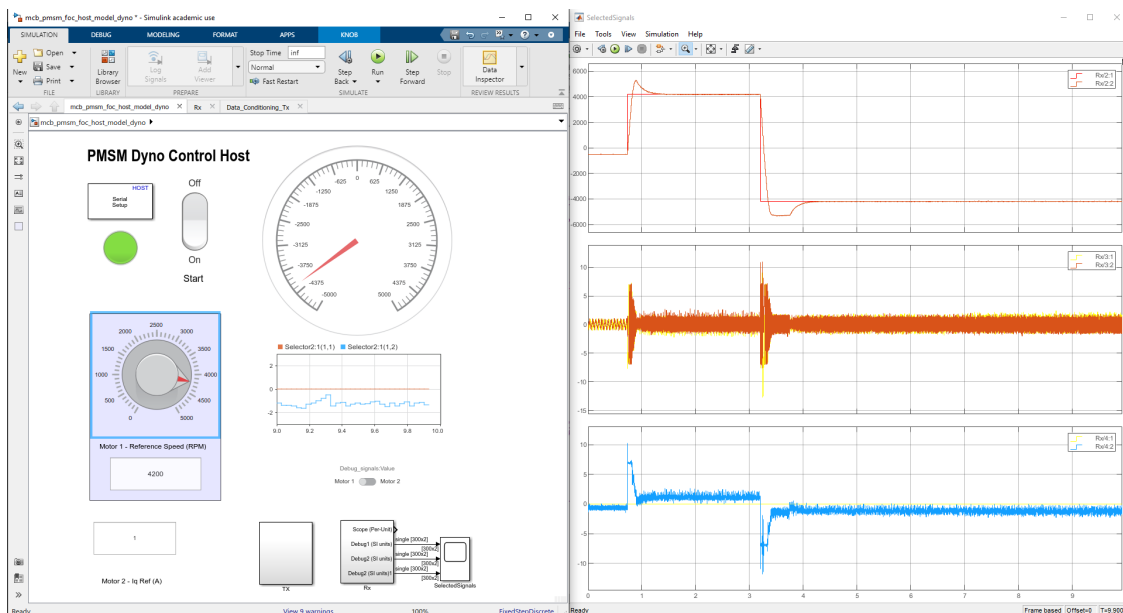


Fig. 6.2.0.2: Interfaz de control para el banco de pruebas

6.3. Obtención de datos y resultados experimentales

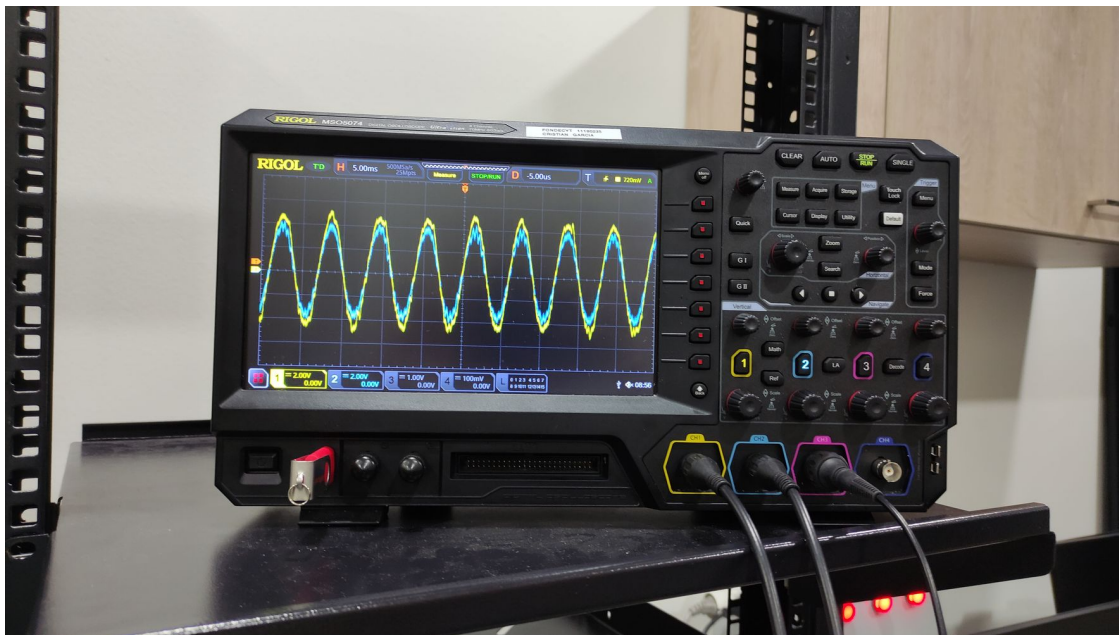


Fig. 6.3.0.1: Osciloscopio digital Rigol, mostrando la corriente del motor controlado en celeste y la corriente del motor de carga en amarillo.

Se logró implementar de forma correcta dos métodos de control, FOC y MPC clásico, sin embargo, MPC de horizonte lejano no funcionó correctamente de forma experimental. No se logró un buen control de velocidad y la máquina eléctrica presentaba bajo torque, al aplicar un poco de presión sobre el eje con la mano se detenía el giro, por lo que no se pudo extraer datos comparables con los otros métodos de control.

Al analizar el código generado de forma automática por el compilador de Simulink, se detectó que al transformar los bloques de código escritos en lenguaje Matlab a código C, que es el lenguaje que se utiliza para programar el microcontrolador, se genera una gran cantidad de líneas y funciones redundantes que aumentan considerablemente el tiempo de ejecución del programa. Evitando que el microcontrolador funcione a la velocidad necesaria para completar los cálculos requeridos por el modelo de predicción.

Por esto, se optó por programar el código en un bloque de lenguaje C directamente dentro del ambiente Simulink, esto permite un balance entre las ventajas de la configuración automática del hardware, como puertos de salida, ADCs, PWMs, asignación de memoria y otras configuraciones repetitivas, y la generación directa del código de control en código C, ya que, el interpretador no debe cambiar de lenguaje al compilar el programa.

Con esto se logró optimizar de forma sustancial el código, y reducir el tiempo de ejecución en un 60%, sin embargo aun así no se logró la implementación de los métodos de control. Por motivos desconocidos, el programa escrito en C no funciona como debería al realizar la selección de los vectores de actuación óptimos, se sospecha que el bloque de código C en Simulink requiere de otros métodos de declaración de variables para su correcto funcionamiento.

En última instancia, se decidió utilizar el código escrito en lenguaje Matlab funcional, pero limitándose a implementar los programas con los periodos de muestreo que sí son posibles de ejecutar en el tiempo asignado.

A continuación se muestran las corrientes de los 2 métodos implementados, usando distintos tiempos de muestreo para el control.

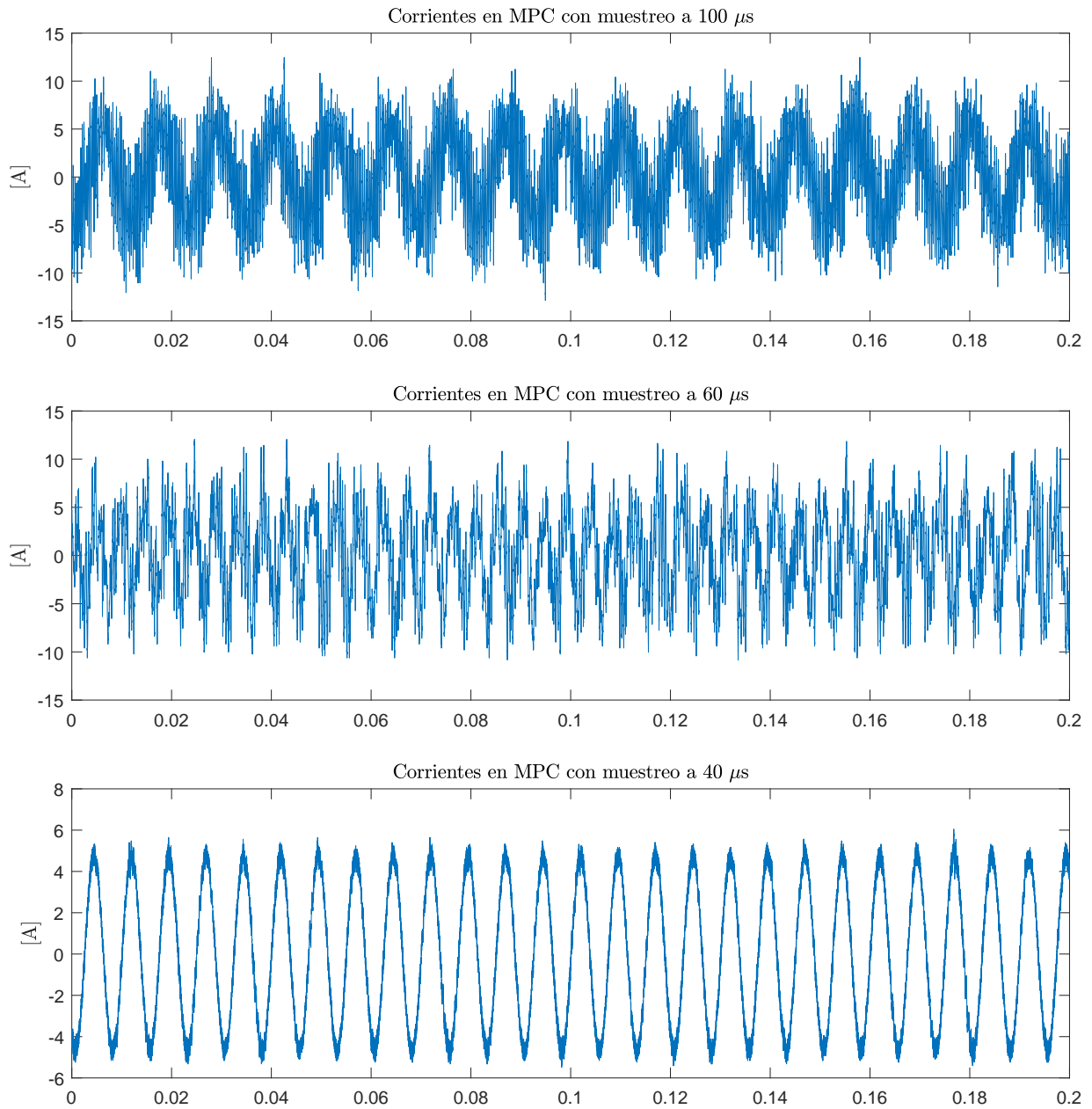


Fig. 6.3.0.2: Corrientes de fase del motor controlado mediante MPC, con diferentes tiempos de muestreo.

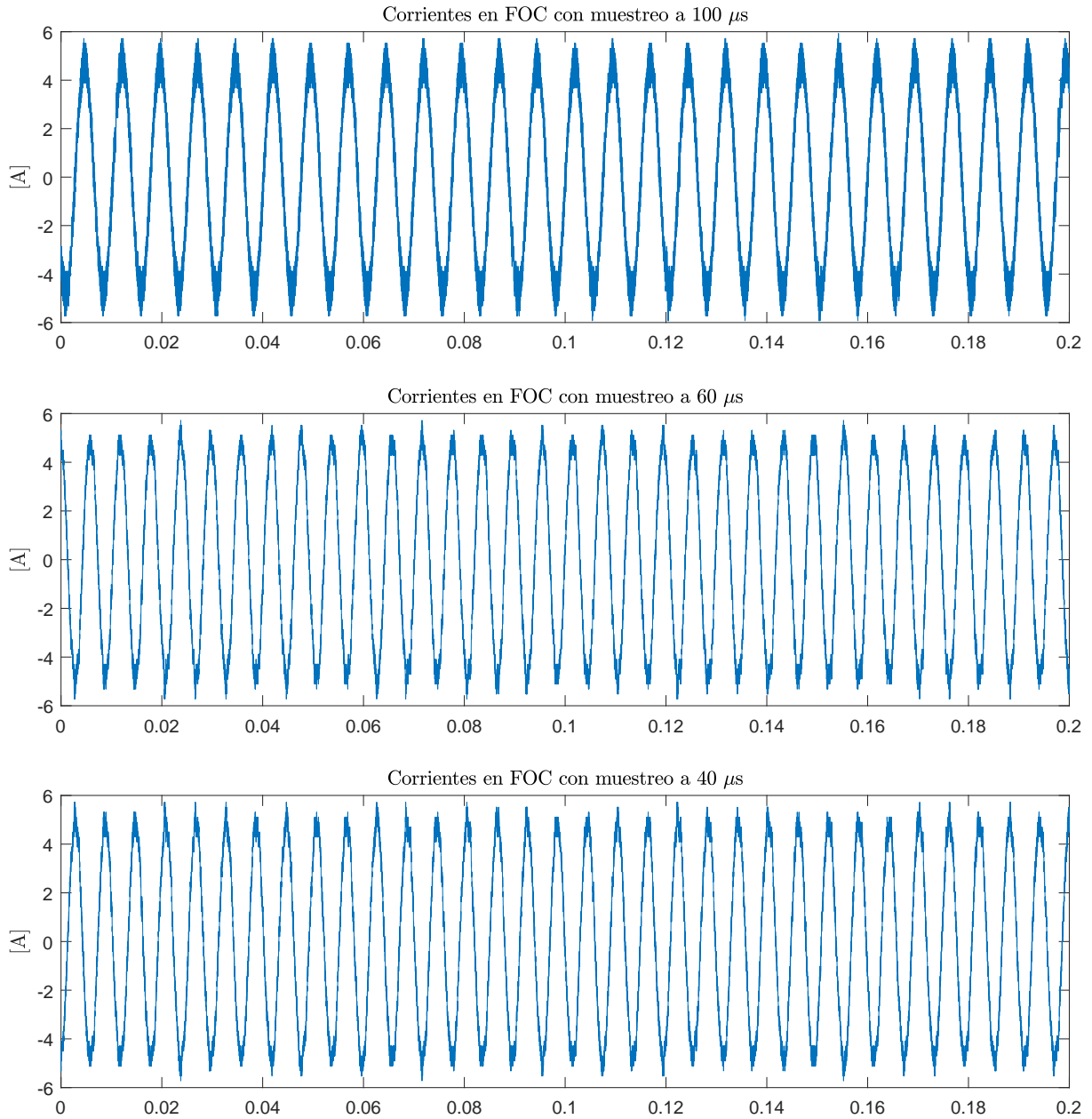


Fig. 6.3.0.3: Corrientes de fase del motor controlado mediante FOC, con diferentes tiempos de muestreo.

Al realizar un cálculo de distorsión armónica total de cada una de las corrientes que se registraron en el osciloscopio digital, se pueden obtener los valores de la distorsión armónica total de la tabla 6.3.0.1.

Tiempo (μs)	THD FOC	THD MPC
40	18,2 %	10,1 %
60	21,2 %	15,8 %
80	25,4 %	20,2 %
100	26,1 %	22,8 %
120	30,5 %	24,2 %

Tabla 6.3.0.1: Tabla de comparación de valores de distorsión armónica total frente a distintos valores de periodo de muestreo.

Además, se gráfico el espectro armónico de los dos métodos de control frente a un mismo periodo de muestreo de $20 \mu s$. Las figuras 6.3.0.4 y 6.3.0.5, muestran estos espectros armónicos hasta una frecuencia de 150 KHz.

Se puede apreciar que el espectro del método FOC presenta armónicos mas definidos que en el caso del control con MPC, el cual presenta un espectro armónico mas distribuido, pero con algunos picos de frecuencia de mayor amplitud, sin embargo, como se mostró en la tabla comparativa anterior, la distorsión armónica total de MPC es menor en todos los casos que la de FOC.

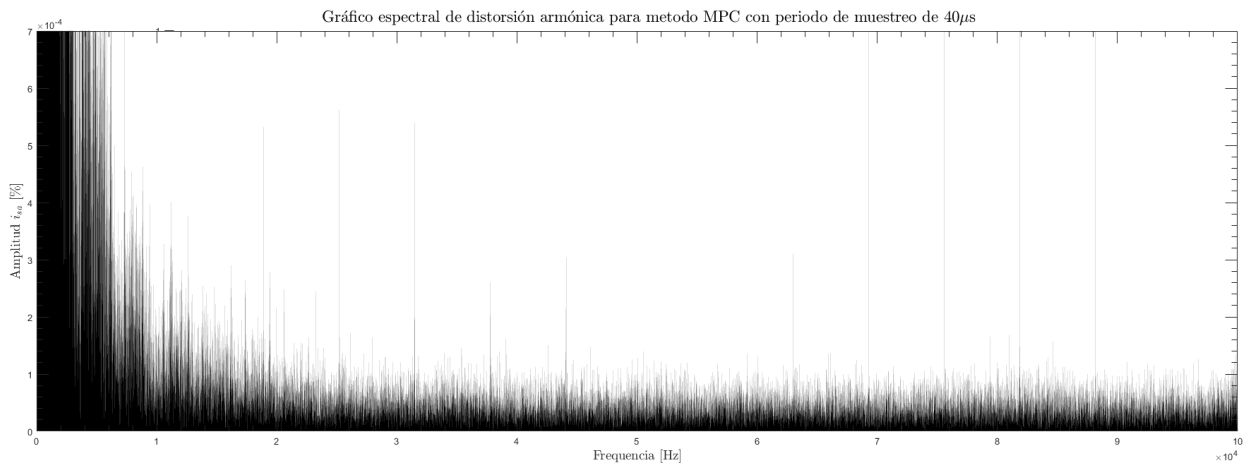


Fig. 6.3.0.4: Espectro armónico de la corriente con MPC a $40 \mu s$

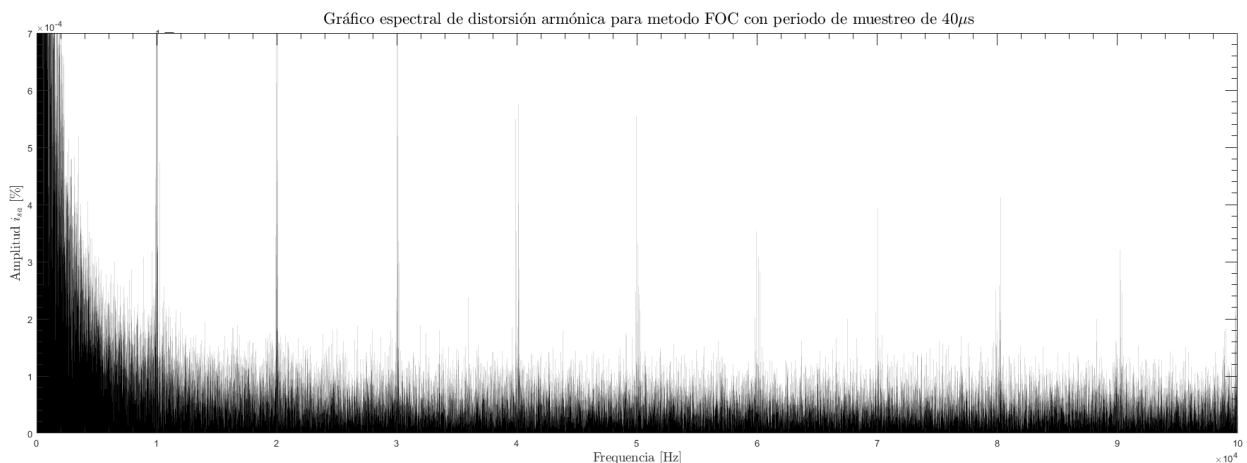


Fig. 6.3.0.5: Espectro armónico de la corriente con FOC a $40 \mu s$

7. Conclusión

Luego de analizar diversos artículos acerca de control de motores síncronos, se pudo observar una tendencia creciente en la aplicación de métodos de control predictivo, en sus diversas variaciones, esto se debe al constante aumento en la capacidad de procesamiento de dispositivos programables, donde se implementa el código de control para los convertidores de potencia.

En este trabajo se simularon tres métodos de control diferentes y se demuestra que los métodos de control predictivo poseen la ventaja que generan una menor distorsión armónica que el control por campo orientado al aplicar una misma frecuencia de conmutación.

La elección de los factores de peso también es de suma importancia y no se puede realizar de forma analítica, ya que depende de muchos factores externos y se deben realizar varias iteraciones para obtener una respuesta deseada.

Por otro lado, al no cumplirse todos los objetivos del proyecto, en particular, la aplicación de LH-MPC de forma experimental, y en consecuencia, la comparación de THD real entre los tres métodos de control propuestos, se puede concluir que la implementación de este método de control requiere de otras estrategias de cálculo, y algoritmos más robustos capaces de procesar la cantidad de operaciones que implica trabajar con este método de control.

Además, la utilización del ambiente Simulink para programar microcontroladores presenta ventajas y desventajas que deben ser analizadas a la hora de implementar un modelo de forma experimental, por un lado, se tiene la simplicidad y rapidez de programar con diagramas de flujo y bloques de comandos, y por otro lado, se pierde la capacidad de ajustar parámetros y optimizar mejor el código generado, ya que, el compilador se encarga de traducir los diagramas de simulink de alto nivel, en otros lenguajes de bajo nivel.

Anexos

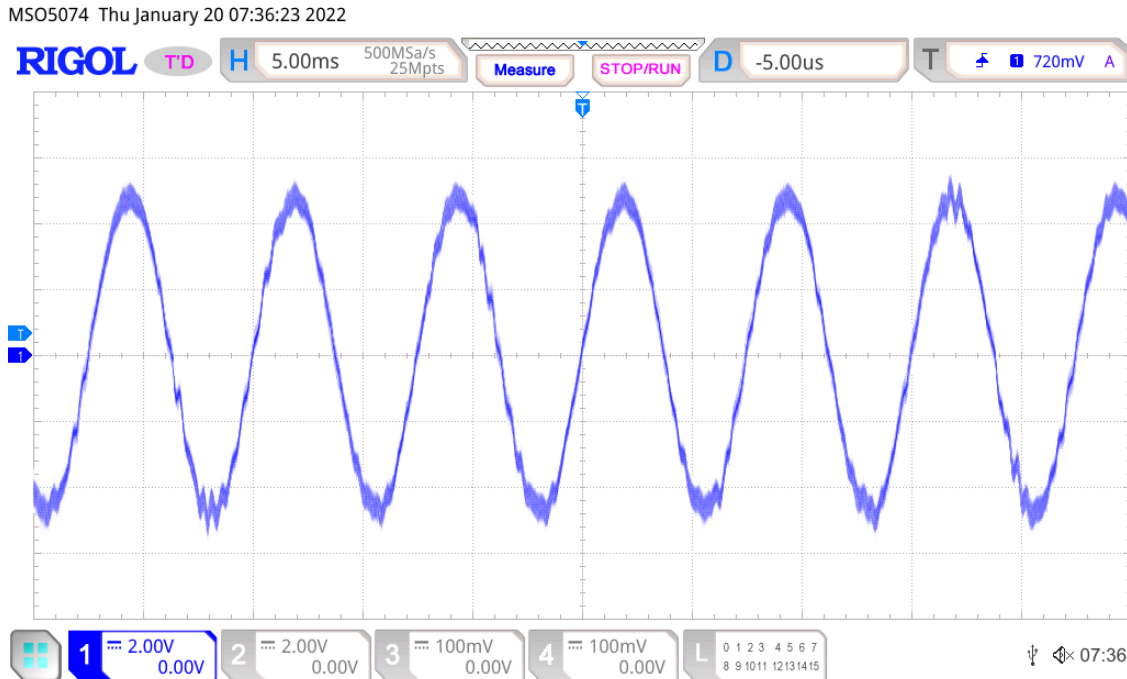


Fig. 7.0.0.1: Captura de pantalla del osciloscopio digital Rigol, mostrando la corriente del motor controlado por MPC. Colores invertidos para mejor claridad.

Listing 1: Código para el control por FOC

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%      Código para metodo FOC      %%
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4  function [Va,Vb,Vc,id,iq,debug] = FOC( ia, ib,spdRef, spdMeas, theta,params)
5
6  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7  %% declaracion de variables
8  persistent errd_l
9  persistent errq_l
10 persistent Id_l
11 persistent Iq_l
12 persistent n
13 if isempty(errd_l)
14     errd_l = single(0);
15 end
16 if isempty(errq_l)
17     errq_l = single(0);
18 end
19 if isempty(Id_l)
20     Id_l = single(0);
21 end
22 if isempty(Iq_l)
23     Iq_l = single(0);
24 end
25 if isempty(n)
26     n = single(1);
27 end
28
29
30 %variables de PI velocidad
31 persistent X_l

```

```

32 persistent X
33 persistent iqref
34 if isempty(X_1)
35     X_1 = single(0);
36 end
37 if isempty(X)
38     X = single(0);
39 end
40 if isempty(iqref)
41     iqref = single(0);
42 end
43
44 %acceder a los parametros del sistema
45
46 KN0m= params.KN0m;
47 KN1m= params.KN1m;
48 KN0e= params.KN0e;
49 KN1e= params.KN1e;
50 ts=    params.Ts;
51
52 idRef= single(0);
53
54 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
55 %% PI Velocidad
56
57 err_spd=spdRef-spdMeas;
58
59 %PI speed with AWU
60 PIAWU_N1m= -(KN1m+KN0m)/(KN0m^2);
61 PIAWU_D1m= -KN1m/KN0m;
62
63 % submuestreo
64 if n == 1
65     iqref=KN0m*(err_spd-X);
66 end
67 if n==10
68     n=single(0);
69 end
70
71 %Limitador de actuacion
72 MAX=single(6);
73 MIN=single(-6);
74
75 iqRef= iqref/(params.kt);
76
77     if iqRef>MAX
78         iqRef=MAX;
79     end
80     if iqRef<MIN
81         iqRef=MIN;
82     end
83
84 % acumulacion AWU
85 X=PIAWU_N1m*iqref+PIAWU_D1m*X_1;
86 X_1=X;
87 n=n+1;
88
89 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
90 %Transformada Clarke
91 iAB=[ia;
92     ib];
93
94 %Clark - Park transformation for current
95 Crk= [ 1      0;
96     0.577  1.155];
97
98 TCrk= [ 1      0;
99     -0.5    0.866;
100     -0.5   -0.866];
101

```

```

102 Prk= [cos(theta) sin(theta);
103        -sin(theta) cos(theta)];
104
105 idq=Prk*Crk*iAB;
106
107 id=idq(1,1);
108 iq=idq(2,1);
109
110 errd=idRef-idq(1,1);
111 errq=iqRef-idq(2,1);
112
113 % PI de velocidad con antiwindup
114 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
115
116 PIAWU_N1e= -(KN1e+KN0e)/(KN0e^2);
117 PIAWU_D1e= -KN1e/KN0e;
118
119 Vd=KN0e*(errd-errd_1);
120 Vq=KN0e*(errq-errq_1);
121
122 MAX=single(1);
123 MIN=single(-1);
124
125 if Vd>MAX
126     Vd=MAX;
127 end
128 if Vd<MIN
129     Vd=MIN;
130 end
131 if Vq>MAX
132     Vq=MAX;
133 end
134 if Vq<MIN
135     Vq=MIN;
136 end
137
138 Vdq=[Vd;
139       Vq];
140
141 %% Transformada inversa
142 Vab= Prk'*Vdq;
143 Vabc= TCrk*Vab;
144
145
146 % variables de salida
147 Va= Vabc(1,1);
148 Vb= Vabc(2,1);
149 Vc= Vabc(3,1);
150
151
152 % debug y persistentes
153 debug=[idRef; id;iqRef;iq;ia;ib];
154 errd_1=PIAWU_N1e*Vd+PIAWU_D1e*Id_1;
155 errq_1=PIAWU_N1e*Vq+PIAWU_D1e*Iq_1;
156 Id_1=errd_1;
157 Iq_1=errq_1;
158
159 end

```

Listing 2: Código para el control por MPC

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%      Código para metodo MPC      %%
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5 function [debug,ialmeas,ialref, SA, SB, SC] = fcn(spdRef,spdMeas,we,iABC, thetaElec,params)
6
7 %% Declaracion de varaibles

```

```

8  %variables de MPC
9
10 persistent old_index n erral errbe weightal weightbe
11 persistent iaBref_old1 iaBref_old2 BEMFaB_old1 BEMFaB_old2
12 if isempty(old_index)
13   old_index = 1;
14 end
15 if isempty(iaBref_old1)
16   iaBref_old1 = zeros(2,1);
17 end
18 if isempty(iaBref_old2)
19   iaBref_old2 = zeros(2,1);
20 end
21 if isempty(BEMFaB_old1)
22   BEMFaB_old1 = zeros(2,1);
23 end
24 if isempty(BEMFaB_old2)
25   BEMFaB_old2 = zeros(2,1);
26 end
27 if isempty(n)
28   n = 1;
29 end
30 if isempty(erral)
31   erral = 0;
32 end
33 if isempty(errbe)
34   errbe = 0;
35 end
36 if isempty(weightal)
37   weightal = 0;
38 end
39 if isempty(weightbe)
40   weightbe = 0;
41 end
42
43 %variables de PI velocidad
44 persistent X_l X Tm
45 if isempty(X_l)
46   X_l = 0;
47 end
48 if isempty(X)
49   X = 0;
50 end
51 if isempty(Tm)
52   Tm = 0;
53 end
54
55 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
56 %% Variable and constants declaration
57
58 Ts=      params.Ts_MPC;
59 KN0m=    params.KN0mpc;
60 KN1m=    params.KN1mpc;
61 kV=      params.kV;
62 Kt=      params.kt;
63 I=       params.mtrxI;
64 A=       params.mtrxA;
65 B=       params.mtrxB;
66 F=       params.mtrxF;
67 imax=    7;
68 lambda=  0.0;
69 x_opt=   1;
70 gopt=    inf;
71 idref=   0;
72
73 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
74 %% PI Speed
75
76 err_spd=spdRef-spdMeas;
77

```



```

78 %PI speed with AWU
79 PIAWU_N1m= -(KN1m+KN0m)/(KN0m^2);
80 PIAWU_D1m= -KN1m/KN0m;
81
82 % submuestreo
83 if n == 1
84     Tm=KN0m*(err_spd-X);
85 end
86
87 %Limitador de actuacion
88 MAX= imax;
89 MIN=-imax;
90
91 iqref = Tm/Kt;
92
93     if iqref>MAX
94         iqref=MAX;
95     end
96     if iqref<MIN
97         iqref=MIN;
98     end
99
100 idqref=[idref;
101         iqref];
102
103 % acumulacion AWU
104 X=PIAWU_N1m*Tm+PIAWU_D1m*X_1;
105 X_1=X;
106
107     if n==10
108         n=0;
109     end
110
111 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
112 %% MPC Current
113
114 %Clark - Park transformation for current
115 Crk= [1   -0.5   -0.5;
116       0   0.866  -0.866];
117
118 Prk= [cos(thetaElec) sin(thetaElec);
119       -sin(thetaElec) cos(thetaElec)];
120
121 BEMFaB=[sin(thetaElec);
122         cos(thetaElec)];
123
124 iaBref= Prk'*idqref;
125 iaB=    0.667*Crk*iaB;
126 idq=   Prk*iaB;
127
128 ialmeas=iaB(1,1);
129 ialref=iaBref(1,1);
130
131 % Value extrapolation
132 BEMFaBk1= 3*BEMFaB- 3*BEMFaB_old1+ BEMFaB_old2;
133 iaBrefk2= 6*iaBref- 8*iaBref_old1+ 3*iaBref_old2;
134
135 %% Estimated output voltage
136
137 % Switching States
138 S =[0 1 1 0 0 0 1 1;
139     0 0 1 1 1 0 0 1;
140     0 0 0 0 1 1 1 1];
141
142 %Delay compensation for voltage vector at k+1
143 V= kV*Crk*S(:,old_index);
144
145 %Clark - Park transformation for k+2
146 V1= kV*Crk*S;
147

```

```

148 %Discrete model matrices using Forward Euler
149 Ad= I+Ts*(A);
150 Bd= Ts*B;
151 Fd= Ts*F;
152
153 %Discrete model matrices using Modified Taylor
154 % Ad= I+Ts*A +(Ts^2/2)*A*A;
155 % Bd= Ts*B +(Ts^2/2)*A*B;
156 % Fd= Ts*F;
157
158 %% Estimated current for k+1
159 iaB1= Ad*iaB + Bd*V+ Fd*BEMFaB*we;
160
161 for i=1:8 %Prediction for k+2
162
163     iaB2=Ad*iaB1 + Bd*V1(:,i) + Fd*BEMFaBk1*we;
164
165     %cost function calculation
166     gk2=(iaBrefk2(1,1)-iaB2(1,1))^2 + (iaBrefk2(2,1)-iaB2(2,1))^2 + lambda*( ...
        S(1,i)-S(1,old.index) + S(2,i)-S(2,old.index) + S(3,i)-S(3,old.index));
167
168     % Hard constraints of stator current
169     if abs(iaB2(1,1)) > imax
170         gk2 = inf;
171     end
172
173     %optimization
174     if(gk2<gopt)
175         gopt=gk2;
176         x_opt=i;
177         x_opt=round(x_opt);
178     end
179
180 end % k+2 end
181
182 index= x_opt;
183
184 %% State change minimizer
185
186 if (index==1 || index==8)
187     if (old.index == 2 || old.index == 4 || old.index == 6)
188         index=1;
189     end
190     if (old.index == 3 || old.index == 5 || old.index == 7)
191         index=8;
192     end
193 end
194
195 %% output switching states
196
197 SA = S(1,index);
198 SB = S(2,index);
199 SC = S(3,index);
200
201 debug=[SA;idq(2,1);iqref];
202
203 %% Save current values
204 old.index = index;
205 n=n+1;
206
207 iaBref_old1=iaBref;
208 iaBref_old2=iaBref_old1;
209 BEMFaB_old1=BEMFaB;
210 BEMFaB_old2=BEMFaB_old1;
211
212 end

```

Listing 3: Código para el control por LH-MPC

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%      Codigo para metodo LH-MPC      %%
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5  function [id,idref,iq,iqref, SA, SB, SC] = fcn(spdRef,spdMeas,we,iABC, thetaElec,params)
6
7  %variables de MPC
8  persistent old_index n erral errbe weightal weightbe
9  persistent iaBref_old1 iaBref_old2 BEMFaB_old1 BEMFaB_old2
10 if isempty(old_index)
11 old_index = 1;
12 end
13 if isempty(iaBref_old1)
14 iaBref_old1 = zeros(2,1);
15 end
16 if isempty(iaBref_old2)
17 iaBref_old2 = zeros(2,1);
18 end
19 if isempty(BEMFaB_old1)
20 BEMFaB_old1 = zeros(2,1);
21 end
22 if isempty(BEMFaB_old2)
23 BEMFaB_old2 = zeros(2,1);
24 end
25 if isempty(n)
26 n = 1;
27 end
28 if isempty(erral)
29 erral = 0;
30 end
31 if isempty(errbe)
32 errbe = 0;
33 end
34 if isempty(weightal)
35 weightal = 0;
36 end
37 if isempty(weightbe)
38 weightbe = 0;
39 end
40
41 %variables de PI velocidad
42 persistent X_l X Tm
43 if isempty(X_l)
44     X_l = 0;
45 end
46 if isempty(X)
47     X = 0;
48 end
49 if isempty(Tm)
50     Tm = 0;
51 end
52
53 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
54 %% Variable and constants declaration
55 %% Model values
56
57 Ts=      params.Ts_MPC;
58 KN0m=    params.KN0mpc;
59 KN1m=    params.KN1mpc;
60 Kt=      params.kt;
61 I=       params.mtrxI;
62 A=       params.mtrxA;
63 B=       params.mtrxB;
64 F=       params.mtrxF;
65 kV=      params.kV;
66 lambda=  params.weight;
67 imax=    7.1;
68 goptlh = 1e10;

```

```

69 x_optlh=1;
70 idref= 0;
71
72 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
73 %% PI Speed
74
75 err_spd=spdRef-spdMeas;
76
77 %PI speed with AWU
78 PIAWU_N1m= -(KN1m+KN0m)/(KN0m^2);
79 PIAWU_D1m= -KN1m/KN0m;
80
81 % submuestreo
82 if n == 1
83     Tm=KN0m*(err_spd-X);
84 end
85
86 %Limitador de actuacion
87 MAX= imax;
88 MIN=-imax;
89
90 iqref= Tm/Kt;
91
92     if iqref>MAX
93         iqref=MAX;
94     end
95     if iqref<MIN
96         iqref=MIN;
97     end
98
99 idqref=[idref;
100         iqref];
101
102 % acumulacion AWU
103 X=PIAWU_N1m*Tm+PIAWU_D1m*X_1;
104 X_1=X;
105
106     if n==10
107         n=0;
108     end
109
110 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
111 %% MPC Current
112
113 %Clark - Park transformation for current
114 Crk= [1   -0.5   -0.5;
115        0   0.866  -0.866];
116
117 Prk= [cos(thetaElec) sin(thetaElec);
118        -sin(thetaElec) cos(thetaElec)];
119
120 iaB= 0.667*Crk*iaABC;
121 idq= Prk*iaB;
122
123 id=idq(1,1);
124 iq=idq(2,1);
125
126 %Reference extrapolation
127 iaBref= Prk'*idqref;
128
129 iaBrefk1= 3*iaBref- 3*iaBref_old1+ iaBref_old2;
130 iaBrefk2= 6*iaBref- 8*iaBref_old1+ 3*iaBref_old2;
131 iaBrefk3= 10*iaBref-15*iaBref_old1+ 6*iaBref_old2;
132 iaBrefk4= 15*iaBref-24*iaBref_old1+10*iaBref_old2;
133
134
135 %Back EMF Extrapolation
136 BEMFaB=[sin(thetaElec);
137          cos(thetaElec)];
138

```

```

139 BEMFaBk1= 3*BEMFaB- 3*BEMFaB_old1+ BEMFaB_old2;
140 BEMFaBk2= 6*BEMFaB- 8*BEMFaB_old1+ 3*BEMFaB_old2;
141 BEMFaBk3= 10*BEMFaB-15*BEMFaB_old1+ 6*BEMFaB_old2;
142 BEMFaBk4= 15*BEMFaB-24*BEMFaB_old1+10*BEMFaB_old2;
143
144 ialref=iaBref(1,1);
145 iberef=iaBref(2,1);
146 ialmeas=iaB(1,1);
147 ibemeas=iaB(2,1);
148
149 %% Estimated output voltage
150 % Switching States
151 S =[0 1 1 0 0 0 1 1;
152     0 0 1 1 1 0 0 1;
153     0 0 0 0 1 1 1 1];
154
155 %Delay compensation for voltage vector at k+1
156 V= kV*Crk*S(:,old_index);
157
158 %Clark - Park transformation for k+2
159 V1= kV*Crk*S;
160
161 %Discrete model matrices using Forward Euler
162 Ad= I+Ts*A;
163 Bd= Ts*B;
164 Fd= Ts*F;
165
166 %Discrete model matrices using Modified Taylor
167 % Ad= I+Ts*A +(Ts^2/2)*A*A;
168 % Bd= Ts*B +(Ts^2/2)*A*B;
169 % Fd= Ts*F;
170
171 %% Estimated current for k+1
172
173 iaB1= Ad*iaB + Bd*V + Fd*BEMFaB*we;
174
175 for i=1:8 %Para k+2
176
177 iaB2= Ad*iaB1 + Bd*V1(:,i) + Fd*BEMFaBk1*we;
178
179 gk2=(iaBrefk2(1,1)-iaB2(1,1))^2 + (iaBrefk2(2,1)-iaB2(2,1))^2 + lambda*(abs( ...
180     S(1,i)-S(1,old_index)) + abs(S(2,i)-S(2,old_index)) + abs(S(3,i)-S(3,old_index)));
181
182 for j=1:8 %Para k+3
183
184 iaB3= Ad*iaB2 + Bd*V1(:,j) + Fd*BEMFaBk2*we;
185
186 gk3=(iaBrefk3(1,1)-iaB3(1,1))^2 + (iaBrefk3(2,1)-iaB3(2,1))^2 + lambda*( ...
187     abs(S(1,j)-S(1,i)) + abs(S(2,j)-S(2,i)) + abs(S(3,j)-S(3,i)));
188
189 for k=1:8 %Para k+4
190
191 iaB4= Ad*iaB3 + Bd*V1(:,k) + Fd*BEMFaBk3*we;
192
193 gk4=(iaBrefk4(1,1)-iaB4(1,1))^2 + (iaBrefk4(2,1)-iaB4(2,1))^2 + lambda*( ...
194     abs(S(1,k)-S(1,j)) + abs(S(2,k)-S(2,j)) + abs(S(3,k)-S(3,j)));
195
196 %Hard constraints of stator current
197 if abs(iaB4(1,1)) > imax
198     gk4 = inf;
199 end
200
201 glh= gk2+gk3+gk4;
202
203 %optimization
204 if(glh<goptlh)
205     goptlh=glh;
206     x.optlh=i;
207 end
208
209

```

```
206         end % k+4 end
207
208     end % k+3 end
209
210 end % k+2 end
211
212 index= x.optlh;
213
214 %% State change minimizer
215
216 if (index==1 || index==8)
217     if (old_index == 2 || old_index == 4 || old_index == 6)
218         index=1;
219     end
220     if (old_index == 3 || old_index == 5 || old_index == 7)
221         index=8;
222     end
223 end
224
225 %% output switching states
226
227 SA = S(1,index);
228 SB = S(2,index);
229 SC = S(3,index);
230
231 %% Save current values
232 old_index = index;
233 n=n+1;
234
235 iaBref_old1=iaBref;
236 iaBref_old2=iaBref_old1;
237 BEMFaB_old1=BEMFaB;
238 BEMFaB_old2=BEMFaB_old1;
239
240 end
```

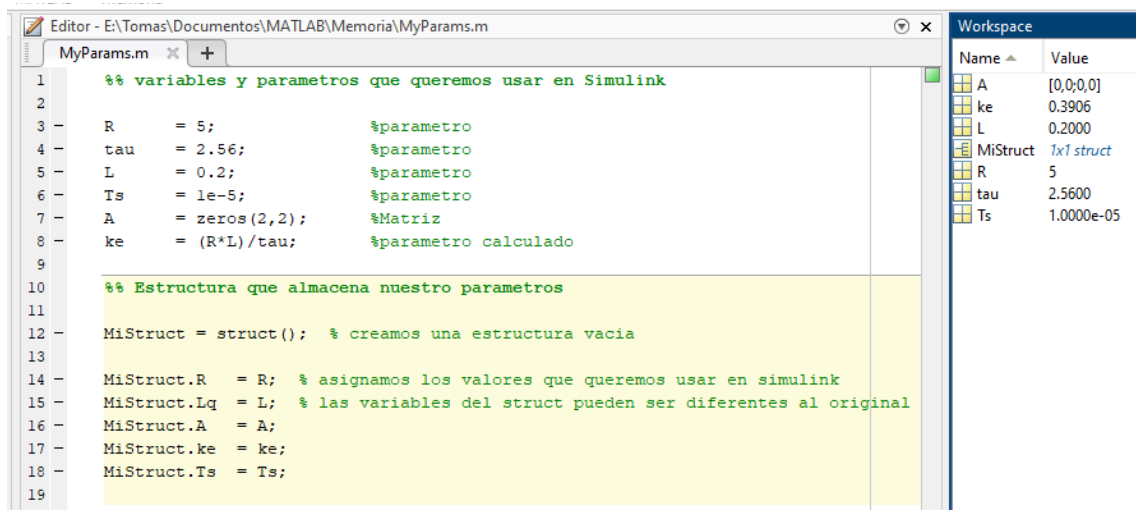
Tutorial para enlazar datos del Workspace de Matlab con funciones de Simulink

José Tomás Pérez

Paso 1

Lo primero que debemos hacer es crear un Script.m que se ejecute 1 vez y que contenga todos los cálculos y parámetros necesarios para ser utilizados en nuestra simulación. En mi caso lo llamé "MyParams.m" y lo guardé en la misma carpeta de mi Simulacion.slx

Una vez que tenemos nuestras variables declaradas, creamos una estructura (línea 12) que almacene todos los datos necesarios para trabajar en la simulación (líneas 14-18), la forma de asignar datos a una estructura se hace de la misma forma que se crea un objeto en otros lenguajes de programación, usando un punto para separar el nombre de la estructura y el nombre de la variable.



The screenshot shows the MATLAB Editor window with the file 'MyParams.m' open. The script contains the following code:

```
1 %% variables y parametros que queremos usar en Simulink
2
3 R = 5; %parametro
4 tau = 2.56; %parametro
5 L = 0.2; %parametro
6 Ts = 1e-5; %parametro
7 A = zeros(2,2); %Matriz
8 ke = (R*L)/tau; %parametro calculado
9
10 %% Estructura que almacena nuestro parametros
11
12 MiStruct = struct(); % creamos una estructura vacia
13
14 MiStruct.R = R; % asignamos los valores que queremos usar en simulink
15 MiStruct.Lq = L; % las variables del struct pueden ser diferentes al original
16 MiStruct.A = A;
17 MiStruct.ke = ke;
18 MiStruct.Ts = Ts;
19
```

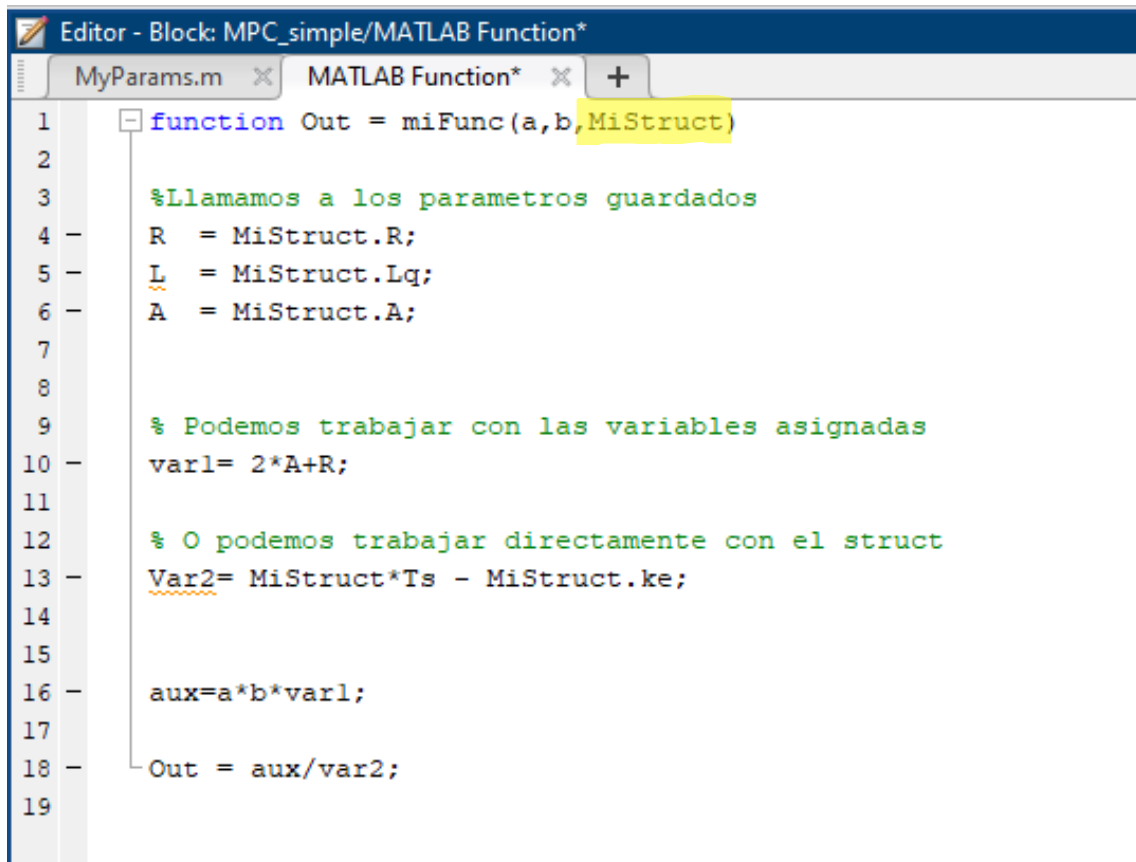
The Workspace window on the right shows the following variables and their values:

Name	Value
A	[0,0,0]
ke	0.3906
L	0.2000
MiStruct	1x1 struct
R	5
tau	2.5600
Ts	1.0000e-05

Debemos asegurarnos que este script se ejecute cada vez que modificamos alguna variable, para actualizar los datos de la estructura en el workspace.

Paso 2

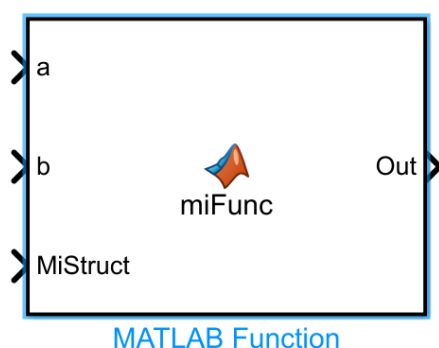
Ahora creamos nuestra función de Matlab en Simulink, y le asignamos como uno de los argumentos, la estructura que creamos en el workspace, siendo cuidadosos con las mayúsculas.



```
Editor - Block: MPC_simple/MATLAB Function*
MyParams.m  MATLAB Function*  +
1  function Out = miFunc(a,b,MiStruct)
2
3  %Llamamos a los parametros guardados
4  R = MiStruct.R;
5  L = MiStruct.Lq;
6  A = MiStruct.A;
7
8
9  % Podemos trabajar con las variables asignadas
10 var1= 2*A+R;
11
12 % O podemos trabajar directamente con el struct
13 Var2= MiStruct*Ts - MiStruct.ke;
14
15
16 aux=a*b*var1;
17
18 Out = aux/var2;
19
```

Como recomendación, es bueno asignar nuevas variables al inicio de nuestra función para ser mas ordenado y saber que variables estamos utilizando, sin embargo, podemos trabajar directamente con los parámetros guardados en la estructura, como muestra la línea 13.

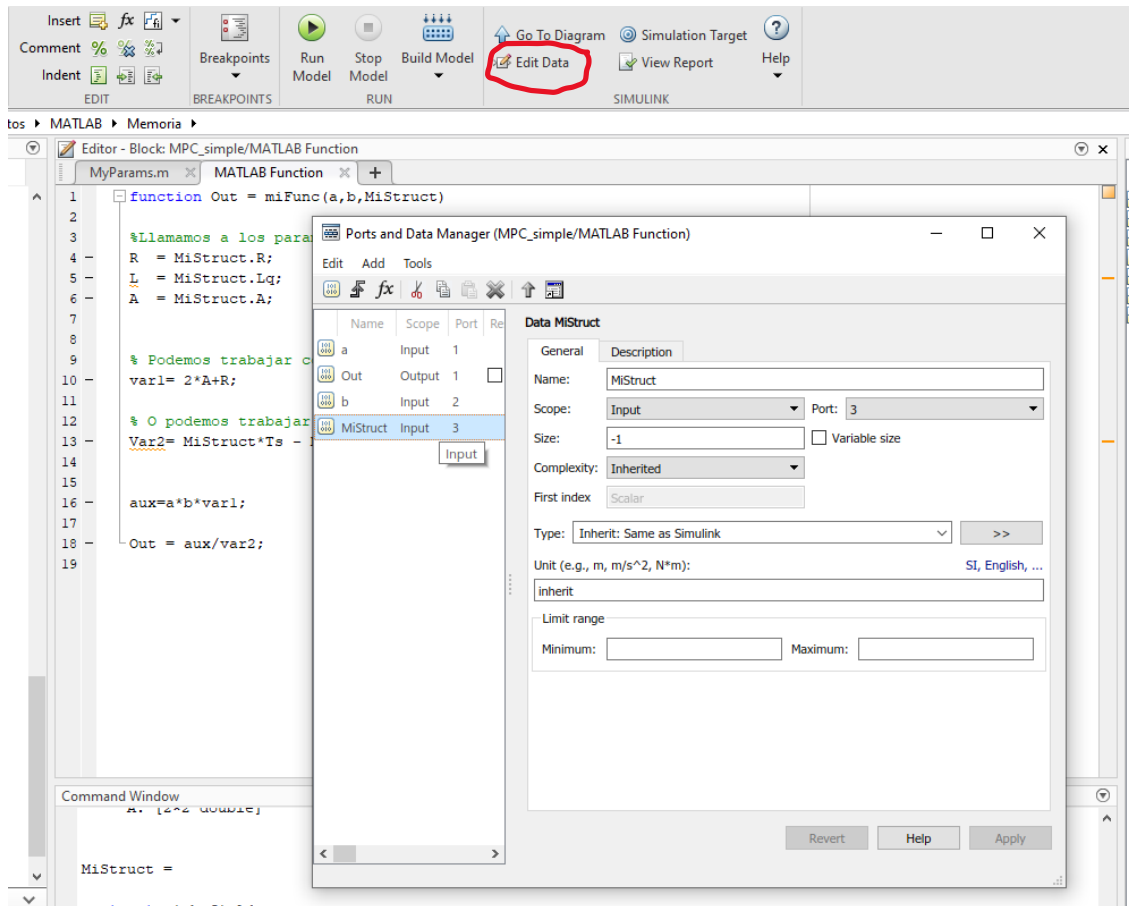
Una vez escrita nuestra función debería verse así dentro de Simulink:



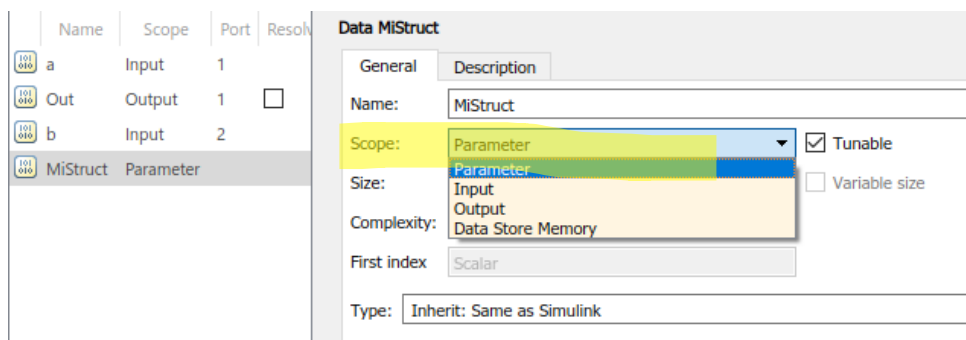
Paso 3

La idea de trabajar con la estructura es que Simulink pueda acceder a los parámetros sin tener que asignarlos como entradas de un bloque. Para esto convertiremos el argumento “MiStruct” en parámetros internos del bloque.

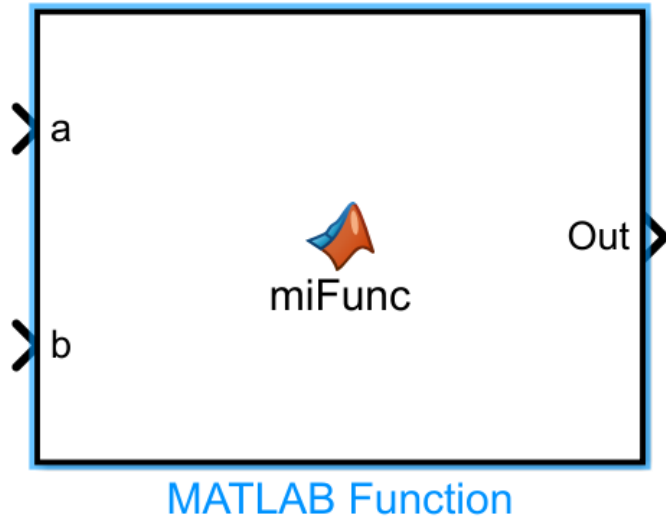
Esto se logra seleccionando la opción “Edit Data” (encerrado en rojo) en la barra de herramientas del editor. Al hacer clic se debería abrir una ventana llamada “Ports and Data Manager”, en esta ventana se ven todos los datos que utiliza nuestra función.



Actualmente los datos de MiStruct son considerados como entradas del bloque. Para cambiarlos a parámetros internos, seleccionamos la sección que dice “Scope” (destacado), y cambiamos “Input” por “Parameter”



Una vez que cambiamos esta opción, la entrada del bloque para MiStruct debería desaparecer, y ahora Simulink almacena las variable y parámetros internamente en el bloque.



Paso 4

Profit!!

Referencias

- [1] Tesla, *Tesla Model S specifications*. 2021. dirección: https://www.tesla.com/es_mx/models?redirect=no.
- [2] Autofacts, *Los 10 mejores autos de alto rendimiento en el mercado chileno*. 2021. dirección: <https://www.autofact.cl/blog/comprar-auto/caracteristicas/autos-rendidores>.
- [3] Y. A. Alamoudi, A. Ferrah, R. Panduranga, A. Althobaiti y F. Mulolani, "State-of-the Art Electrical Machines for Modern Electric Vehicles," en *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, 2019, págs. 1-8. DOI: 10.1109/ICASET.2019.8714343.
- [4] F. Wang, S. Li, X. Mei, W. Xie, J. Rodríguez y R. M. Kennel, "Model-Based Predictive Direct Control Strategies for Electrical Drives: An Experimental Evaluation of PTC and PCC Methods," *IEEE Transactions on Industrial Informatics*, vol. 11, n.º 3, págs. 671-681, 2015. DOI: 10.1109/TII.2015.2423154.
- [5] M. S. Trivedi y R. K. Keshri, "Evaluation of Predictive Current Control Techniques for PM BLDC Motor in Stationary Plane," *IEEE Access*, 2020. DOI: test.
- [6] A. G. de Castro, W. C. A. Pereira, T. E. P. Almeida, C. M. R. de Oliveira, J. R. B. A. Monteiro y A. A. de Oliveira, "Improved finite control-set model-based direct power control of BLDC motor with reduced torque ripple," en *2016 12th IEEE International Conference on Industry Applications (INDUSCON)*, 2016, págs. 1-6. DOI: 10.1109/INDUSCON.2016.7874530.
- [7] M. Jeong, S. Fuchs y J. Biela, "When FPGAs Meet Regionless Explicit MPC: An Implementation of Long-horizon Linear MPC for Power Electronic Systems," en *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020, págs. 3085-3092. DOI: 10.1109/IECON43393.2020.9254277.
- [8] T. Dorfling, H. du Toit Mouton, T. Geyer y P. Karamanakos, "Long-Horizon Finite-Control-Set Model Predictive Control With Nonrecursive Sphere Decoding on an FPGA," *IEEE Transactions on Power Electronics*, vol. 35, n.º 7, págs. 7520-7531, 2020. DOI: 10.1109/TPEL.2019.2956213.
- [9] B. Long, T. Cao, W. Fang, K. T. Chong y J. M. Guerrero, "Model Predictive Control of a Three-Phase Two-Level Four-Leg Grid-Connected Converter Based on Sphere Decoding Method," *IEEE Transactions on Power Electronics*, vol. 36, n.º 2, págs. 2283-2297, 2021. DOI: 10.1109/TPEL.2020.3006432.
- [10] L. Ortombina, P. Karamanakos y M. Zigliotto, "Robustness Analysis of Long-Horizon Direct Model Predictive Control: Permanent Magnet Synchronous Motor Drives," en *2020 IEEE 21st Workshop on Control and Modeling for Power Electronics (COMPEL)*, 2020, págs. 1-8. DOI: 10.1109/COMPEL49091.2020.9265855.
- [11] A. Andersson y T. Thiringer, "Assessment of an Improved Finite Control Set Model Predictive Current Controller for Automotive Propulsion Applications," *IEEE Transactions on Industrial Electronics*, vol. 67, n.º 1, págs. 91-100, 2020. DOI: 10.1109/TIE.2019.2898603.
- [12] P. Acuna, C. A. Rojas, R. Baidya, R. P. Aguilera y J. E. Fletcher, "On the Impact of Transients on Multistep Model Predictive Control for Medium-Voltage Drives," *IEEE Transactions on Power Electronics*, vol. 34, n.º 9, págs. 8342-8355, 2019. DOI: 10.1109/TPEL.2018.2889565.
- [13] V. S. Conte, T. Geyer y W. van der Merwe, *Model Predictive Control for MV Drives*. 2019. dirección: <https://new.abb.com/drives/>.
- [14] I. Takahashi y T. Noguchi, "A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor," *IEEE Transactions on Industry Applications*, vol. IA-22, n.º 5, págs. 820-827, 1986. DOI: 10.1109/TIA.1986.4504799.
- [15] Y. Tang y G. Lin, "Direct torque control of induction motor based on self-adaptive PI controller," en *2010 5th International Conference on Computer Science Education*, 2010, págs. 1230-1234. DOI: 10.1109/ICCSE.2010.5593448.
- [16] P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo y J. Rodriguez, "Predictive Control in Power Electronics and Drives," *IEEE Transactions on Industrial Electronics*, vol. 55, n.º 12, págs. 4312-4324, 2008. DOI: 10.1109/TIE.2008.2007480.

- [17] J. Rodríguez, M. P. Kazmierkowski, J. R. Espinoza, P. Zanchetta, H. Abu-Rub, H. A. Young y C. A. Rojas, "State of the Art of Finite Control Set Model Predictive Control in Power Electronics," *IEEE Transactions on Industrial Informatics*, vol. 9, n.º 2, págs. 1003-1016, 2013. DOI: 10.1109/TII.2012.2221469.
- [18] G. Bode, P. C. Loh, M. Newman y D. Holmes, "An improved robust predictive current regulation algorithm," *IEEE Transactions on Industry Applications*, vol. 41, n.º 6, págs. 1720-1733, 2005. DOI: 10.1109/TIA.2005.858324.