

UNIVERSIDADE DE SÃO PAULO–USP
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

**Guilherme Hiroshigue Motomura
Hashimoto**

**CONTROLE DE VELOCIDADE DE
MOTORES BRUSHLESS
UTILIZANDO PLATAFORMA ARM
CORTEX M3 E M4**

São Carlos
2014

**Guilherme Hiroshigue Motomura
Hashimoto**

**CONTROLE DE VELOCIDADE DE
MOTORES BRUSHLESS
UTILIZANDO PLATAFORMA ARM
CORTEX M3 E M4**

Trabalho de conclusão de Curso apresentado à Escola
de Engenharia de São Carlos, da Universidade de São
Paulo

Curso de Engenharia Elétrica com ênfase em Eletrônica

Orientador: José Roberto Boffino de Almeida Monteiro

São Carlos

2014

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

H344c Hashimoto, Guilherme Hiroshigue Motomura
CONTROLE DE VELOCIDADE DE MOTORES BRUSHLESS
UTILIZANDO PLATAFORMA ARM CORTES M3 E M4 / Guilherme
Hiroshigue Motomura Hashimoto; orientador José Roberto
Boffino de Almeida Monteiro. São Carlos, 2014.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2014.

1. Motor Brushless. 2. Inversor de Frequência. 3.
Controle de Velocidade. 4. ARM Cortex M. I. Título.

FOLHA DE APROVAÇÃO

Nome: Guilherme Hiroshigue Motomura Hashimoto

Título: "Controle de velocidade de motores Brushless utilizando Plataforma ARM Cortex M3 e M4"

Trabalho de Conclusão de Curso defendido e aprovado
em 27/06/2014,

com NOTA 10,0 (dez, zero), pela Comissão Julgadora:

Prof. Dr. José Roberto Boffino de Almeida Monteiro - (Orientador - SEL/EESC/USP)

Prof. Dr. Osmar Ogashawara - (UFSCar)

Mestre Marcelo Patricio de Santana - (Doutorando - SEL/EESC/USP)

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Homero Schiabel

Dedico esse trabalho aos meus pais
Paulo Yoshinori Hashimoto e Chieko Motomura Hashimoto

Agradecimentos

Ao Prof. Dr. José Roberto Boffino de Almeida Monteiro pela sua orientação e auxílio que foram essenciais no desenvolvimento desse trabalho e a todo o pessoal do LACEP que sempre me ajudou no laboratório.

À minha família são carlense por toda a amizade e companheirismo ao longo dos anos.

À minha família paulistana por todo o amor que me trouxe até aqui.

Resumo

Hashimoto, Guilherme H. M. **CONTROLE DE VELOCIDADE DE MOTORES BRUSHLESS UTILIZANDO PLATAFORMA ARM CORTEX M3 E M4.** 55 p. Trabalho de Conclusão de Curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, 2014.

Este trabalho tem como objetivo realizar a medição e o controle digital de velocidade da máquina *brushless* Siemens 1FT5062-AC01 utilizando um inversor de frequência controlado por uma plataforma baseada em microcontroladores ARM Cortex M3 e M4. Para isso, são estudados a modelagem da máquina *brushless*, seus circuitos de acionamento e métodos de controle. Após isso, são analisados os aspectos de implementação desse controle, no ponto de vista da plataforma de controle digital.

Palavras-chave: motor brushless, inversor de frequência, controle de velocidade, ARM Cortex M.

Abstract

Hashimoto, Guilherme H. M. **BRUSHLESS MOTOR SPEED CONTROL WITH ARM CORTEX M3 AND M4 BASED PLATFORM**. 55 p. Monograph – São Carlos School of Engineering, University of São Paulo, 2014.

This work's goal is to implement digital speed control of the Siemens 1FT5062-AC01 brushless motor using an frequency inverter controlled by an ARM Cortex M3 and M4 based platform. In order to do so, the modeling of the brushless machines, its driving circuits and control methods are studied. Next, the implementation aspects of the speed control are analyzed under the scope of the digital control platform.

Keywords: brushless motor, frequency inverter, speed control, ARM Cortex M.

Lista de ilustrações

Figura 1	Circuito equivalente do motor BLDC trifásico	24
Figura 2	Sinais de três sensores Hall e as tensões nas fases correspondentes . . .	25
Figura 3	Exemplo de um encoder incremental, retirado de http://www.orientalmotor.com	26
Figura 4	Sinais do encoder e suas sequências de rotação	26
Figura 5	Sinais de PWM na operação em 1 quadrante. Amarelo: chave positiva da fase A, Verde: chave negativa da fase A, Roxo: chave positiva da fase B, Rosa: chave negativa da fase B.	28
Figura 6	Sinais de PWM na operação em 2 quadrantes. Amarelo: chave positiva da fase A, Verde: chave negativa da fase A, Roxo: chave positiva da fase B, Rosa: chave negativa da fase B	28
Figura 7	Diagrama de blocos do controlador PI com o método anti wind up proposto em (MONTEIRO et al., 2013)	33
Figura 8	Diagrama de blocos da malha fechada de corrente	33
Figura 9	Diagrama de blocos da malha fechada de velocidade	34
Figura 10	Diagrama de blocos das malha de corrente e velocidade	34
Figura 11	Plataforma construída utilizada no controle de velocidade do motor brushless. A placa Headcon está à esquerda e a placa Invslave à direita.	35
Figura 12	Diagrama de blocos das malha de corrente e velocidade	37
Figura 13	Braço do inversor trifásico conectado ao dissipador e à placa Invslave .	38
Figura 14	Funcionamento do modo encoder dos timers da série STM32F4, extraído de (ST Microelectronics, 2014)	41
Figura 15	Amostragem do contador do timer	41
Figura 16	Motor brushless Siemens 1FT5 062 AC01	42
Figura 17	Resposta simulada com $K_{pS} = 0.0277$, $K_{iS} = 0.111$, $K_{pI} = 20.8$, $K_{iI} = 2143$	44
Figura 18	Resposta simulada com $K_{pS} = 8$, $K_{iS} = 2.5$, $K_{pI} = 100$, $K_{iI} = 2500$. .	44

Figura 19	Montagem experimental para a realização dos ensaios	45
Figura 20	Relação entre as tensões induzidas e os sensores Hall	46
Figura 21	Relação entre as tensões induzidas e os sensores Hall	46
Figura 22	Medida de T_{hall}	47
Figura 23	Medida de velocidade obtida através do GDB	47
Figura 24	Resposta ao degrau de $\omega_{ref} = 60rad/s$ experimental e simulada	48
Figura 25	Resposta da velocidade e corrente ao degrau $\omega_{ref} = 60rad/s$	49
Figura 26	Resposta da velocidade e corrente ao sinal ω_{ref}	50
Figura 27	Resposta da velocidade e corrente ao sinal ω_{ref}	50
Figura 28	Período de execução das tarefas de comunicação e controle no micro- controlador mestre.	51

Lista de tabelas

Tabela 1	Tabela de comutação das fases	27
Tabela 2	Parâmetros do IFR420	37
Tabela 3	Mensagem enviada pelo mestre	39
Tabela 4	Modo de operação das fases	39
Tabela 5	Mensagem enviada pelo escravo	40
Tabela 6	Parâmetros do motor Siemens 1FT5 062 AC01	42

Lista de siglas

ARM Advanced RISC Machine

AD Analógico-Digital

BLDC Brushless Direct Current

CC Corrente Contínua

DSP Digital Signal Processor

EESC Escola de Engenharia de São Carlos

FOSS Free Open Source Software

GCC GNU Compiler Collection

GDB GNU Debugger

GNU GNU's not Unix

IGBT Insulated Gate Bipolar Transistor

LACEP Laboratório de Controle e Eletrônica de Potência

LED Light Emitting Diode

PI Proporcional Integrativo

PID Proporcional Integrativo Derivativo

PWM Pulse Width Modulation

RAM Random Access Memory

SPI Serial Peripheral Interface

USB Universal Serial Bus

Sumário

1	Introdução	21
2	Motores BLDC	23
2.1	Equações elétricas	23
2.2	Equações mecânicas	24
2.3	Sensores de posição e velocidade	25
2.4	Acionamento elétrico	27
3	Controle	31
3.1	Controlador PI digital	31
3.2	Controle de corrente e velocidade	33
4	Materiais e métodos	35
4.1	Plataforma de controle	35
4.1.1	Topologia da plataforma de controle	36
4.1.2	Ambiente de desenvolvimento	36
4.1.3	Ponte inversora	37
4.1.4	Protocolo SPI	38
4.1.5	Medição de velocidade do rotor	40
4.2	Motor BLDC	42
4.3	Funcionamento do programa	43
4.4	Sintonia dos controladores	43
5	Testes e resultados	45
5.1	Sincronização dos sensores Hall e fases do motor	45
5.2	Medição de velocidade	47
5.3	Ensaio em malha fechada	48
5.4	Tempo de execução do programa	48

Conclusão	53
Referências	55

Introdução

Desde sua introdução no século XIX, as máquinas de corrente contínua (CC) têm um grande histórico e importância no desenvolvimento industrial. Devido a sua flexibilidade e versatilidade, elas são utilizadas nas mais variadas aplicações: desde locomotivas elétricas de vários kW, até pequenos aparelhos eletrodomésticos. Essa versatilidade se deve muito ao fato de que sabe-se controlar com relativa facilidade o torque, a velocidade e posição da máquina (FITZGERALD, 1975).

Dentre as máquinas CC, a máquina de corrente contínua sem escovas, ou do inglês *brushless direct current motor* (BLDCM), tem ganhado maior espaço e importância frente aos motores CC convencionais comutados por escovas, principalmente em áreas como automação industrial, eletrodomésticos, aeronáutica, medicina, robótica, entre outras.

As principais vantagens do BLDCM em relação às máquinas convencionais são maior eficiência energética, menor peso e volume, menor custo de manutenção, ausência de centelhamento e seu conseqüente ruído. Por outro lado, as máquinas *brushless* são em geral mais caras e tem controle mais complexo. Assim, para cada tipo de aplicação, esses fatores devem ser considerados na hora da escolha do tipo de motor a ser utilizado (YEDAMALE, 2003).

Uma importante aplicação dos motores *brushless* é na tração de veículos híbridos e/ou elétricos. Tendo em vista a preocupação global na redução da emissão de gases estufa, veículos de alta eficiência energética têm ganhado mais investimentos tanto por parte da iniciativa pública como da privada. Assim, dominar o controle de velocidade dos motores *brushless* é um assunto de grande interesse.

Existem diversas estratégias modernas de controle de velocidade que já foram estudadas e propostas, tais como controle ótimo, adaptativo e não linear. No entanto, essas metodologias têm base teórica complexa e são de difícil implementação (IBRAHIM; HASSAN; SHOMER, 2013).

Em (GAMAZO-REAL, 2010) o autor apresenta uma revisão técnica de métodos de aquisição de velocidade usando diversos tipos de sensores e técnicas *sensorless*. Apesar das técnicas *sensorless* conseguirem reduzir o custo do sistema, elas requerem algoritmos

mais complexos.

Por fim, estuda-se o controle PID que ao melhorar as respostas de transiente e regime permanente, já oferece uma solução satisfatória, simples e eficiente para muitos dos casos de aplicações práticas (ANG; CHONG; LI, 2005).

Um outro aspecto importante de estudo é o hardware utilizado no controle e acionamento de motores brushless. Dentre as diversas possibilidades, se destacam o uso de microprocessadores e FPGA's, do inglês, *Field Programmable Gate Array*. A escolha de hardware depende dos objetivos específicos de cada aplicação e também do custo final do sistema.

Este trabalho tem por objetivo implementar a medição de velocidade do rotor de um motor BLDC usando um encoder óptico e realizar também o controle de velocidade da máquina. Para isso, utilizou-se uma plataforma de controle baseada em microcontroladores ARM Cortex M3 e M4, apresentada no capítulo 4.

Motores BLDC

Os motores BLDC são motores síncronos de imã permanente de fluxo trapezoidal. Assim, o campo magnético do rotor gira com a mesma frequência do campo girante gerado no estator, que possui tensão induzida de formato trapezoidal. Diferentemente dos motores CC com escova, a comutação do estator é feita eletronicamente, por meio do chaveamento de suas fases.

Nesse capítulo serão apresentadas as equações matemáticas que regem o funcionamento desse tipo de motor, assim como os circuitos utilizados em seu acionamento, considerando apenas as máquinas trifásicas por serem as mais comuns.

2.1 Equações elétricas

O circuito elétrico equivalente do estator é apresentado na Figura 1. Esse circuito é constituído de três fases simétricas conectadas em Y. Cada fase possui uma resistência de enrolamento, uma indutância que considera a auto-indutância dos enrolamentos de uma fase e a indutância mútua entre os outros enrolamentos da máquina, e a tensão interna do motor que representa a força eletromotriz trapezoidal induzida pela variação do fluxo do rotor no estator.

A partir desse circuito, as tensões nos terminais das fases do estator são dadas por (MONTEIRO, 1997):

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = R_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + (L_s - M_s) \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} + \begin{bmatrix} V_n \\ V_n \\ V_n \end{bmatrix} \quad (1)$$

Onde:

V_a, V_b, V_c : tensões nos terminais das fases do estator

i_a, i_b, i_c : correntes nas fases do estator

e_a, e_b, e_c : tensões induzidas no estator

R_s : resistência dos enrolamentos das fases

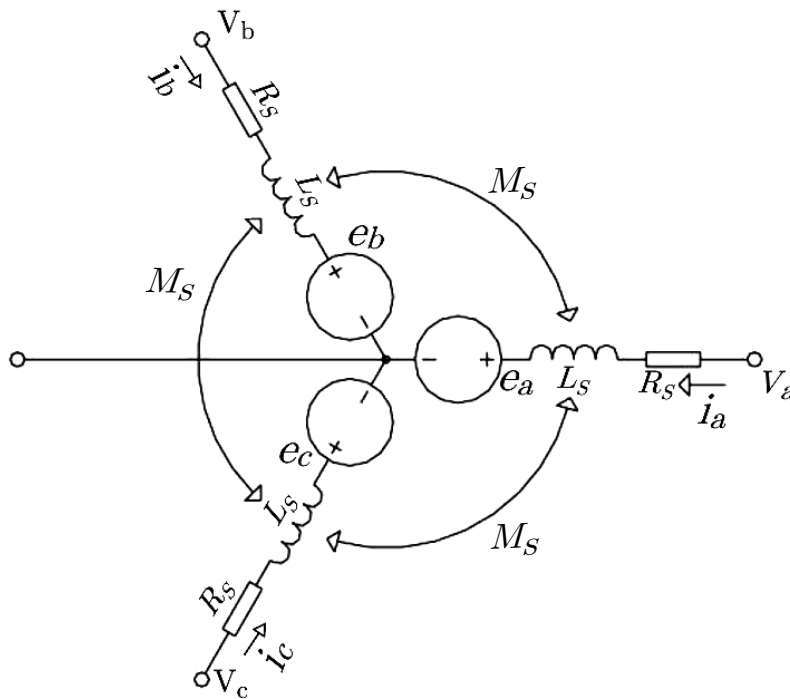


Figura 1 – Circuito equivalente do motor BLDC trifásico

L_s : auto-indutância dos enrolamentos das fases

M_s : indutância mútua entre os enrolamentos das fases

V_n : tensão de neutro

2.2 Equações mecânicas

A equação mecânica do rotor é dada por uma equação diferencial de primeira ordem que relaciona o torque aplicado com o momento de inércia e coeficiente de atrito viscoso da máquina e sua carga:

$$T_{el} = J \frac{d\omega_m}{dt} + B\omega_m + T_c \quad (2)$$

Onde:

T_{el} : torque eletromagnético produzido no rotor

T_c : torque aplicado pela carga

J : momento de inércia do rotor e da carga

B : constante de atrito viscoso do rotor e da carga

ω_m : velocidade angular mecânica do rotor

As máquinas *brushless* podem variar quantos ao número de pares de polos magnéticos (n_{pp}). Assim, a velocidade angular em radianos elétricos por segundo ω_{el} é relacionada com a velocidade angular mecânica ω_m por meio da expressão:

$$\omega_{el} = n_{pp}\omega_m \quad (3)$$

2.3 Sensores de posição e velocidade

Como a comutação dos motores BLDC é controlada eletronicamente, é necessário conhecer a posição angular do rotor para que as fases do estator possam ser acionadas corretamente. Essa posição pode ser detectada utilizando sensores de efeito Hall.

Os sensores Hall produzem nível lógico alto ou baixo indicando se um polo norte ou sul dos ímãs do rotor está passando pelo sensor em determinado momento. Em velocidades constantes, cada sensor produz um sinal de onda quadrada.

Os motores BLDC trifásicos possuem três sensores Hall alinhados dentro do estator de forma que os sensores estejam defasados em 120° elétricos. A combinação dos sinais dos três sensores permite dividir cada ciclo elétrico em seis setores, ou seja, fornece uma resolução de 60° elétricos, como apresentado na Figura 2:

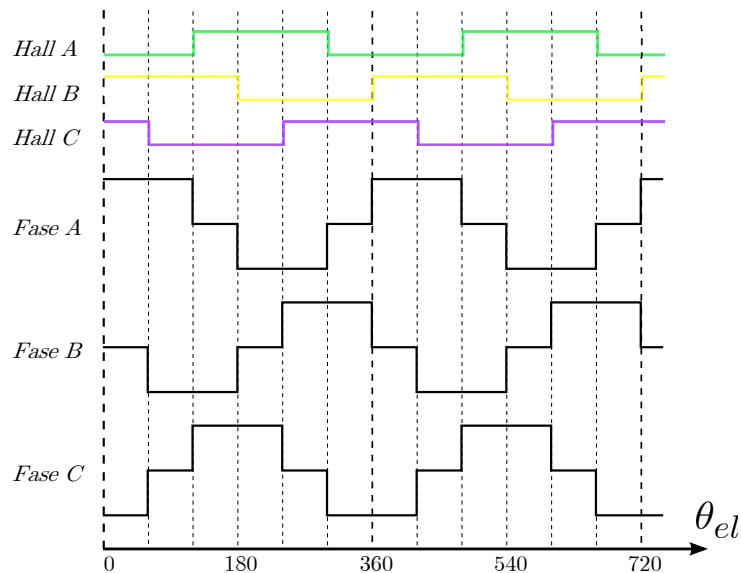


Figura 2 – Sinais de três sensores Hall e as tensões nas fases correspondentes

Os sensores Hall também podem ser utilizados para a medição da velocidade do rotor. Uma maneira de se fazer isso é contar o número de transições positivas de um dos sensores em um período de amostragem T_w . A cada n_{pp} subidas de borda, temos uma volta completa do rotor.

Porém, quando deseja-se obter a velocidade com maior precisão e em um período de amostragem menor da velocidade, faz-se o uso de encoders.

Um tipo muito comum de encoder é o encoder óptico. Nesse tipo de dispositivo, acopla-se um disco com furos igualmente espaçados em volta do eixo do rotor. Alinhados com os furos estão um emissor e um receptor de luz, tipicamente um LED e um fotodiodo. Assim, cada vez que um furo se alinha com o par óptico, um pulso de tensão é produzido nos terminais do foto diodo. É comum encontrar discos com mais de 1000 furos, o que fornece uma resolução de velocidade muito maior em relação aos sensores hall.

Para determinar também o sentido de rotação, usam-se encoders incrementais, que possuem dois fotorreceptores alinhados de tal forma que os pulsos produzidos pelos receptores estejam defasados de 90° , como ilustrado na Figura 3. Assim, a combinação do sinal dos dois receptores e sua sequência permitem determinar o sentido de rotação. Na Figura 4 podemos observar a diferença de fase dos dois canais. Um atraso ou avanço do canal B em relação ao canal A indica sentido horário ou anti-horário de rotação.

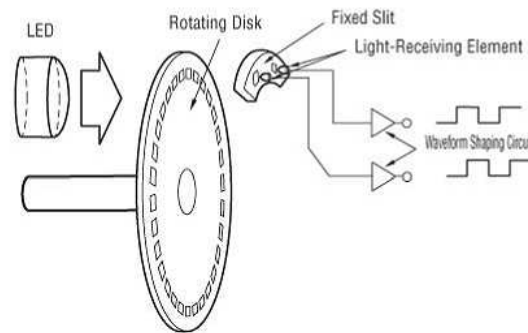


Figura 3 – Exemplo de um encoder incremental, retirado de <http://www.orientalmotor.com>

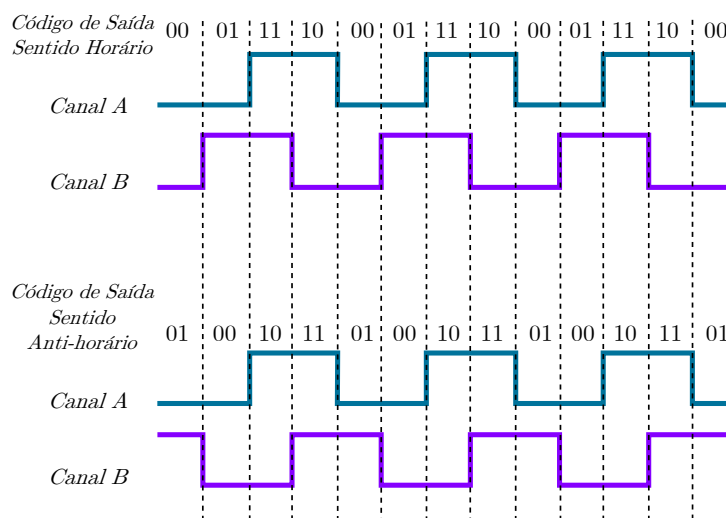


Figura 4 – Sinais do encoder e suas sequências de rotação

2.4 Acionamento elétrico

O acionamento de motores BLDC trifásicos é feito utilizando pontes inversoras trifásicas, onde cada fase do motor é conectada a um braço do inversor. O método de acionamento mais comum é o modo 6 pulsos (six step) 120°.

Nesse método, apenas duas chaves do inversor se encontram em condução simultânea, sendo uma chave positiva e uma negativa. Como resultado, cada chave conduz por 120° elétricos, de forma que a cada 60° elétricos, duas fases estarão ligadas e a outra fase estará desligada.

Associando esse método de acionamento com os sinais gerados pelos sensores Hall que também dividem um ciclo elétrico em setores de 60° elétricos, é possível obter uma lógica simples de comutação das fases, como a apresentada na tabela seguinte:

Tabela 1 – Tabela de comutação das fases

Hall A	Hall B	Hall C	Fase A	Fase B	Fase C
1	1	0	OFF	+	-
0	1	0	-	+	OFF
0	1	1	-	OFF	+
0	0	1	OFF	-	+
1	0	1	+	-	OFF
1	0	0	+	OFF	-

O sentido de direção de rotação pode ser alterado invertendo-se os sinais dos sensores. Isso fará com que a sequência de comutação também seja invertida.

Uma maneira para que se possa controlar a tensão aplicada nas fases, é aplicar a modulação em largura de pulso, ou *pulse width modulation* (PWM), nas chaves do inversor. Para um sinal genérico de PWM defini-se:

$$\delta = \frac{T_{on}}{T_{PWM}} \quad (4)$$

Onde:

δ : Ciclo de trabalho

T_{on} : Período de nível lógico alto

T_{PWM} : Período do sinal de PWM

Existem três modos de aplicar o PWM às chaves, o que fará com que o conversor trabalhe em um, dois ou quatro quadrantes.

Na operação em um quadrante, apenas a chave positiva recebe o sinal de PWM, como ilustrado na Figura 5. Esse modo não permite inverter a tensão na carga, assim como não permite que a carga regenere energia para a fonte.

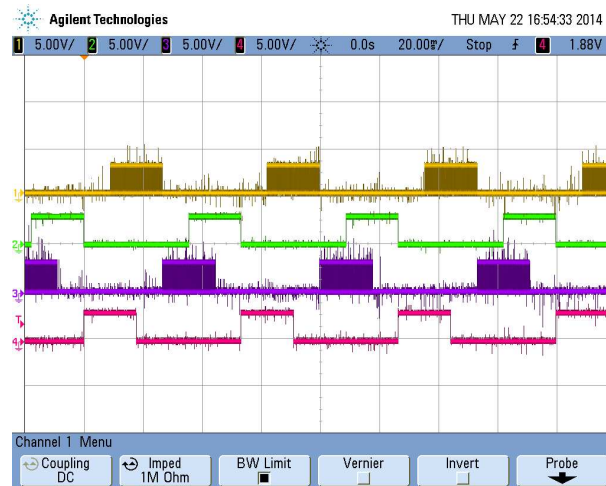


Figura 5 – Sinais de PWM na operação em 1 quadrante. Amarelo: chave positiva da fase A, Verde: chave negativa da fase A, Roxo: chave positiva da fase B, Rosa: chave negativa da fase B.

Já na operação em dois quadrantes, a chave positiva recebe o sinal PWM e a chave negativa recebe o sinal complementar na fase positiva, como ilustrado na Figura 6. Esse modo não permite inverter a tensão na carga, mas permite que a carga regenere energia para a fonte.

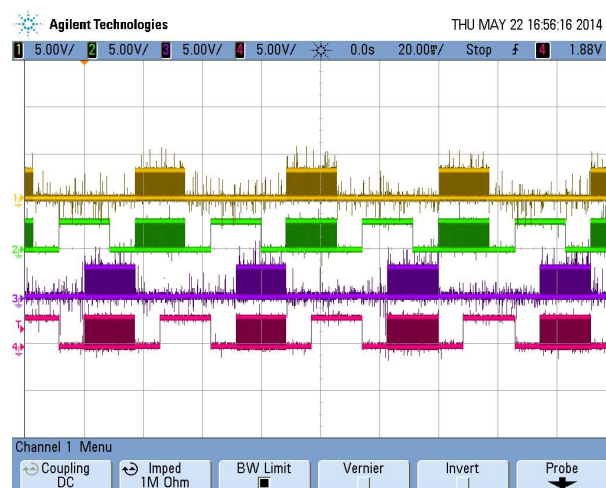


Figura 6 – Sinais de PWM na operação em 2 quadrantes. Amarelo: chave positiva da fase A, Verde: chave negativa da fase A, Roxo: chave positiva da fase B, Rosa: chave negativa da fase B

Nos dois casos, para o modo de condução contínua, a tensão aplicada nas fases é dada por:

$$V_{app} = \delta V_{bus} \quad (5)$$

Onde:

V_{app} : Tensão aplicada à fase

V_{bus} : Tensão do barramento CC

Por fim, na operação em quatro quadrantes, a chave positiva recebe o sinal PWM e a chave negativa recebe o sinal complementar na fase positiva, e na fase negativa, a chave negativa recebe o sinal PWM e a chave positiva recebe o sinal complementar. Esse modo permite tanto a inversão de tensão na carga como a regeneração de energia para a fonte.

Nesse caso, a tensão aplicada nas fases será:

$$V_{app} = (2\delta - 1)V_{bus} \quad (6)$$

Controle

O tipo de controlador utilizado nesse trabalho foi o controlador PI, devido ao seu bom custo-benefício quando consideradas a complexidade de implementação e melhoria da resposta dinâmica do sistemas.

Uma configuração possível é de utilizar apenas um controlador PI que receba o sinal de erro da velocidade do rotor e atue diretamente no valor médio da tensão aplicada as fases do motor. Outra possibilidade, é que um controlador PI receba o erro da velocidade e gere um sinal de corrente, que por sua vez alimenta uma malha de controle PI da corrente.

A última configuração permite saturar a corrente do barramento e a tensão aplicada as fases, ao passo que a primeira permite saturar apenas a tensão. Assim, a fim de proteger o motor de picos de corrente, a configuração com dois controladores é preferível.

Como a malha de controle será implementada em um microprocessador, as técnicas de amostragem e controle digital devem ser consideradas para o desenvolvimento desse tipo de controlador.

3.1 Controlador PI digital

O controlador PID é um dos tipos mais comuns de controladores e possui aplicações nas mais diversas áreas da ciência devido a sua simplicidade de cálculo e implementação.

A equação de um controlador PID pode ser dada pela seguinte expressão:

$$u(t) = K_p e(t) + K_i \int_0^t (e(\tau) d\tau) + K_d \frac{d}{dt} e(t) \quad (7)$$

Onde:

$u(t)$: saída do controlador

$e(t)$: erro entre referência e a variável de interesse

K_p, K_i, K_d : constantes proporcional, integrativa e derivativa respectivamente

Na prática, 80% dos controladores PID não utilizam o termo derivativo. Embora esse termo possa melhorar a resposta dinâmica da planta, o mesmo também é mais sensível a

ruídos de alta frequência que podem comprometer a estabilidade do sistema. Além disso, a contribuição do termo integrativo é mais expressiva em plantas de segunda ordem, do que em plantas de primeira ordem (ANG; CHONG; LI, 2005). Assim, os dois controladores utilizados nesse trabalho serão controladores PI.

Como esses controladores serão implementados em um microprocessador, a equação 7 deve ser discretizada em função de um período de amostragem T_s , e a equação em tempo discreto correspondente é:

$$u_k = u_{k-1} + K_p(e_k - e_{k-1}) + K_i T_s e_k \quad (8)$$

Um problema comum nesse tipo de topologia é o da saturação do termo integrativo devido ao acúmulo excessivo de erro no integrador. Esse tipo de saturação, conhecida em inglês como *wind-up*, leva a um elevado valor de *overshoot* que pode ser prejudicial para o sistema sendo controlado.

Existem diversos métodos *anti wind-up* que visam corrigir esse problema. Em (MONTEIRO et al., 2013), o autor propõe um método onde aplica-se uma função de saturação exclusivamente ao termo integrativo. De forma genérica, a função de saturação pode ser dada por:

$$\text{sat}(x, X_{max}, X_{min}) = \begin{cases} X_{max}, & \text{se } x > X_{max} \\ X_{min}, & \text{se } x < X_{min} \\ x, & \text{se } X_{min} \leq x \leq X_{max} \end{cases} \quad (9)$$

A ação proporcional a_k é definida por:

$$a'_k = K_p e_k \quad (10)$$

$$a_k = \text{sat}(a'_k, A_{max}, A_{min}) \quad (11)$$

Assim, utilizando-se esse método, a equação discreta para o controlador PI é dada por:

$$u'_k = a_k - a_{k-1} + K_i T_s e_k + u_{k-1} \quad (12)$$

$$u_k = \text{sat}(u'_k, U_{max}, U_{min}) \quad (13)$$

O diagrama de blocos desse controlador é apresentado na Figura 7, em que:

$$K'_I = K_i T_s \quad (14)$$

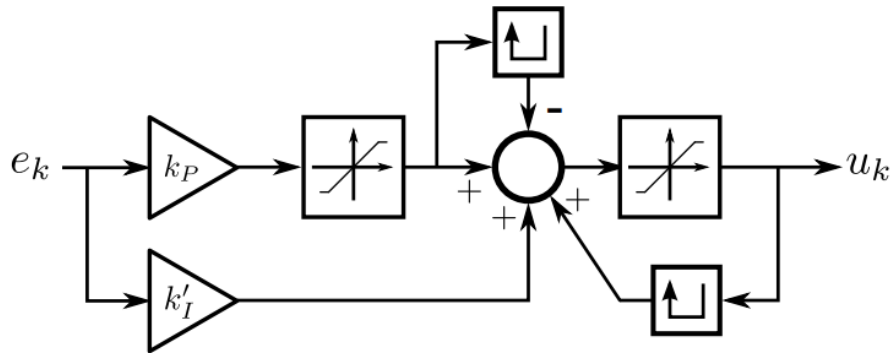


Figura 7 – Diagrama de blocos do controlador PI com o método anti wind up proposto em (MONTEIRO et al., 2013)

3.2 Controle de corrente e velocidade

A partir do modelo do motor apresentado no capítulo 2.1, defini-se a função de transferência entre tensão e corrente de fase como sendo:

$$G_I(s) = \frac{I(s)}{V(s)} = \frac{1}{sL_s + R_s} \quad (15)$$

Assim, o controle em malha fechada da corrente pode ser representado pelo diagrama da figura 8:

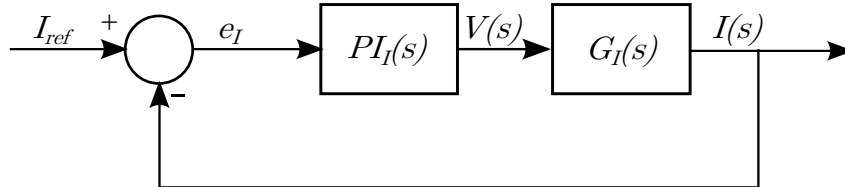


Figura 8 – Diagrama de blocos da malha fechada de corrente

Onde:

I_{ref} : referência de corrente

e_I : erro da corrente

$PI_I(s)$: função de transferência do controlador PI de corrente

Por sua vez, a função de transferência entre torque aplicado à máquina e a velocidade angular do rotor é dada por:

$$G_\omega(s) = \frac{\omega(s)}{T(s)} = \frac{1}{sJ + B} \quad (16)$$

O diagrama de controle em malha fechada do torque velocidade é apresentado na Figura 9.

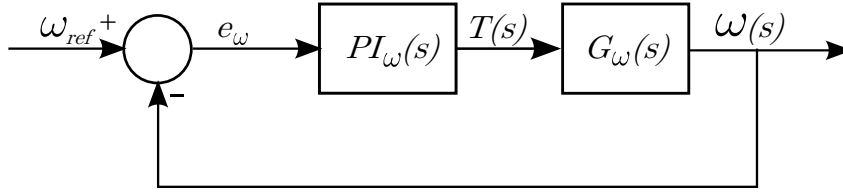


Figura 9 – Diagrama de blocos da malha fechada de velocidade

Onde:

ω_{ref} : referência de velocidade

e_ω : erro da velocidade

$PI_\omega(s)$: função de transferência do controlador PI de velocidade

O torque elétrico e a corrente de fase podem ser relacionadas através da seguinte expressão:

$$T_{el} = K_t I \quad (17)$$

Essa equação permite integrar as duas malhas de controle, pois o sinal de torque gerado pelo controlador PI na malha de velocidade pode ser dividido por K_t para gerar a corrente de referência I_{ref} que alimenta a malha de corrente. Multiplicando a corrente na fase por K_t obtemos novamente o torque aplicado à máquina.

Assim, a associação das duas malhas apresenta a estrutura ilustrada na Figura 10:

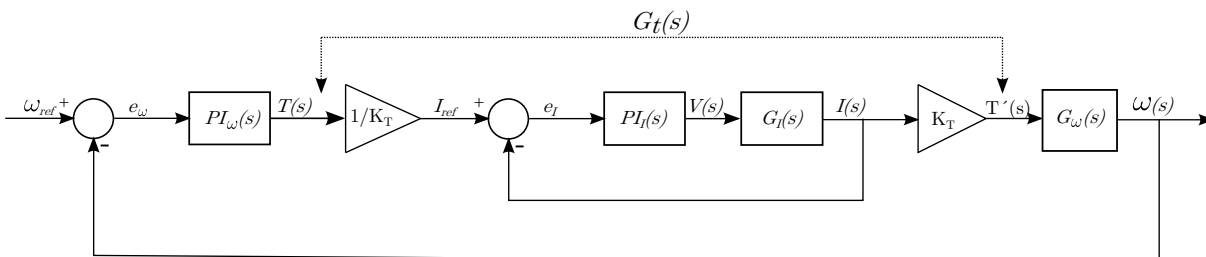


Figura 10 – Diagrama de blocos das malhas de corrente e velocidade

Do ponto de vista do controlador de velocidade, existe agora uma função de transferência $G_t(s)$ entre o sinal de torque gerado $T(s)$ e o torque aplicado à máquina $T'(s)$ que deve ser levada em conta no momento da sintonia do primeiro controlador PI. Caso a dinâmica das grandezas elétricas seja muito mais rápida do que a dinâmica das grandezas mecânicas da máquina, é razoável aproximar $G_t(s) \approx 1$ de modo que $T(s) \approx T'(s)$. Desse modo, cada PI pode ser sintonizado independentemente sem maiores consequências.

Materiais e métodos

Neste capítulo são apresentados a plataforma de controle e o motor utilizados no desenvolvimento desse trabalho. Em seguida é feita a descrição do funcionamento do programa de controle e por fim, o método de sintonia dos controladores PI.

4.1 Plataforma de controle

A plataforma de controle digital utilizada nesse trabalho existente no Laboratório de Controle e Eletrônica de Potência (LACEP), se trata de uma plataforma que utiliza dois tipos de subsistemas: um mestre, que será referenciado por Headcon e outro sistema escravo, aqui denominado por Invslave. Ambos possuem placas de circuito impresso e microcontroladores próprios às funções que desempenham. A plataforma é apresentada na Figura 11. Como esse trabalho foi o primeiro a utilizar essa plataforma, atividades como solda de componentes, depuração de erros da placa e desenvolvimento do firmware básico também foram realizadas.

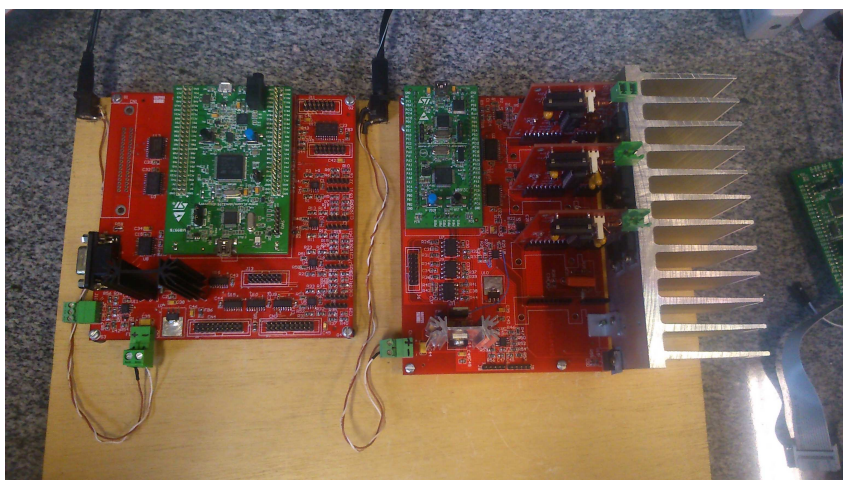


Figura 11 – Plataforma construída utilizada no controle de velocidade do motor brushless. A placa Headcon está à esquerda e a placa Invslave à direita.

4.1.1 Topologia da plataforma de controle

A plataforma foi pensada de forma que pudesse ser utilizada em outros projetos de inversores e/ou outros tipos de conversores de energia chaveados que possam vir a ser desenvolvidos no LACEP. Uma vez que o firmware básico para os dois sistemas é desenvolvido e os seus componentes são testados, futuros projetistas não necessitarão desprender grande quantidade de tempo desenvolvendo e testando seus circuitos, e assim, poderão alocar mais tempo ao desenvolvimento de seus controladores.

No centro do processamento da placa Invslave, está o kit de desenvolvimento da ST Microelectronics o STM32VLDISCOVERY que tem como microcontrolador o STM32F100. Esse microcontrolador tem arquitetura ARM Cortex-M3 de 32 bits, 128 KB de memória Flash, 8 KB de memória RAM e frequência máxima de operação de 24 MHz. Como principais vantagens, pode-se destacar que esse kit já possui integrado a ele o programador USB ST-LINK e conectores que dão acesso a todos os pinos do microcontrolador. Caso algum componente do kit falhe, o mesmo pode ser trocado rapidamente por outro kit, sem que seja necessário fazer ou desfazer qualquer tipo de solda.

A Invslave é responsável pelas conversões AD de tensão das fases, corrente do barramento CC e temperatura do inversor. Além disso, também é o Invslave que gera os sinais de PWM que são aplicados a cada transistor do inversor.

A placa Headcon por sua vez processa os dados utilizando o kit STM32F4DISCOVERY, também da ST Microelectronics. Assim como o STM32VLDISCOVERY, esse kit conta com o programador USB ST-LINK e acesso aos pinos do microcontrolador. Porém, o microcontrolador presente no F4DISCOVERY é o STM32F407, que possui arquitetura ARM Cortex-M4F. As principais diferenças em relação à F100, é que o F407 possui 1 MB de memória Flash, 192 KB de memória RAM, pode operar em até 168 MHz e principalmente, possui instruções DSP e de ponto flutuante. Todas essas vantagens somadas fazem com que o F407 seja mais rápido e eficiente em cálculos mais complexos.

A Headcon é portanto a responsável pelos cálculos de controle e de atuação do sistema. No caso específico da aplicação desse trabalho, os sensores hall e encoders do BLDC também são conectados à placa. Assim, utilizando um protocolo de comunicação SPI, o mestre pode receber os dados do AD do escravo, fazer os cálculos, e enviar os comandos de atuação das fases.

A Figura 12 representa de forma simplificada o esquema de funcionamento do sistema como um todo.

4.1.2 Ambiente de desenvolvimento

Os programas de ambos microcontroladores foram desenvolvidos em linguagem C, num ambiente de desenvolvimento em GNU/Linux. O compilador utilizado foi o ARM-Embedded-GCC (disponível em <https://launchpad.net/gcc-arm-embedded>). Já o desen-

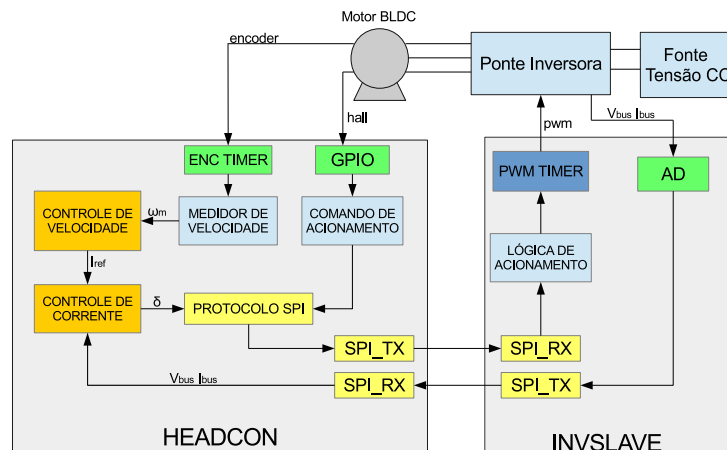


Figura 12 – Diagrama de blocos das malha de corrente e velocidade

volvimento do código assim como sua depuração foram realizados utilizando o GDB (disponível em <http://www.gnu.org/software/gdb/>) como extensão do Emacs23 (disponível em <http://www.gnu.org/software/emacs/>). A conexão entre o programador ST-LINK e o GDB foi feita utilizando o programa OpenOCD (disponível em <http://openocd.sourceforge.net/>). A toolchain como um todo consiste integralmente de programas FOSS (*free open source software*), o que garante baixo custo e flexibilidade para o projeto como um todo.

4.1.3 Ponte inversora

A ponte inversora trifásica utilizada está integrada à placa Invslave. Cada braço do inversor é uma outra placa de circuito impresso que possui dois IGBTs, seus drivers, conversores CC/CC e lógica de proteção. Ainda visando a modularidade e flexibilidade, todas as placas são ligadas ao Invslave utilizando conectores de pino. Assim, caso uma das fases falhe, a mesma pode ser removida independentemente das outras e reposta.

A Figura 13 mostra uma placa de um braço do inversor conectado ao Invslave.

Para esse trabalho, foram usados os IGBT's IRGB420U da International Rectifier. Os principais parâmetros desse transistor estão apresentadas na Tabela 2.

Tabela 2 – Parâmetros do IRF420

$I_{MAX}(25^{\circ}C)$	14A
$V_{CE_{MAX}}$	500V
$V_{CE(SAT)}$	< 3.0V
$V_{GE(TH)}$	5.5V
etc	etc

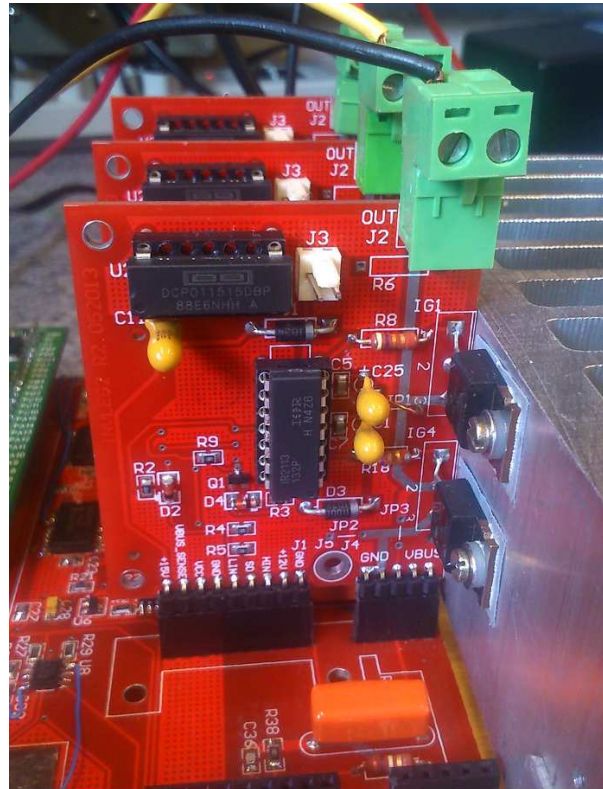


Figura 13 – Braço do inversor trifásico conectado ao dissipador e à placa Invslave

4.1.4 Protocolo SPI

O SPI, do inglês *Serial Peripheral Interface*, é um protocolo de comunicação serial *full duplex* muito utilizado em sistemas microcontrolados. O SPI permite que um mestre se comunique com um ou mais dispositivos escravos em velocidades de até 100 MHz.

Nesta aplicação, um protocolo foi criado para estabelecer a comunicação entre a placa Headcon e a placa Invslave, que desempenham a função de mestre e escravo respectivamente. Em ambos os casos, o tamanho da mensagem a ser transmitida é de 4 palavras de 16 bits cada.

Para o escravo, as quatro palavras a serem enviadas são os resultados das conversões dos canais AD de 12 bits. As conversões a serem enviadas são definidas pelo mestre, uma vez que essa informação está contida na mensagem que o escravo recebe.

No mestre, a primeira palavra da mensagem, além de informar os canais a serem amostrados, também envia o comando que cada fase do inversor deve obedecer. O formato da mensagem gerado pelo mestre é apresentado na Tabela 3:

Tabela 3 – Mensagem enviada pelo mestre

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1.	1	0	0	0	V25	AD1	AD0	C2	C1	C0	B2	B1	B0	A2	A1	A0
2.	δ															
3.	0x0000															
4.	0x0000															

Word 1: Palavra de comando

Bits 15:12 Reserved

Bit 11 V25: Medir variáveis de inicialização

0: O valor de Temp é recebido na 4ª palavra

1: O valor de V25 é recebido na 4ª palavra

Bits 10:9 AD[1:0]: Seleção de VOx

00: VO1

01: VO2

10: VO3

11: VO4

Bits 8:6 C[2:0]: Modo de operação da fase C

Bits 5:3 B[2:0]: Modo de operação da fase B

Bits 2:0 A[2:0]: Modo de operação da fase A

Word 2: Valor do ciclo de trabalho δ

Word 3: Deve ser mantido em 0

Word 4: Deve ser mantido em 0

O modo de operação de cada fase A, B, C é dado pela Tabela 4, onde H e L representam os transistores superiores e inferiores de cada braço do inversor, respectivamente. O número 1 indica que o transistor deve fechar, o número 0 indica que o transistor deve abrir, δ indica que o ciclo de trabalho enviado na 2ª palavra do protocolo deve ser aplicado, e $\bar{\delta}$ indica que o complemento do ciclo de trabalho ($1 - \delta$), deve ser aplicado.

Tabela 4 – Modo de operação das fases

Código	H	L
000	0	0
001	0	1
010	1	0
011	δ	0
100	δ	$\bar{\delta}$
101	$\bar{\delta}$	δ

O escravo por sua vez deve responder com uma mensagem que possui o formato apresentado na Tabela 5.

Tabela 5 – Mensagem enviada pelo escravo

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1.	Vbus															
2.	Ibus															
3.	VOx															
4.	V25/Temp															

Word 1: Valor da tensão de barramento Vbus

Word 2: Valor médio da corrente de barramento Ibus

Word 3: Valor da tensão na fase x, x=1,2,3,4

Word 4: Valor de V25 ou da temperatura do dissipador Temp

4.1.5 Medição de velocidade do rotor

A medição de velocidade do rotor é feita por meio da contagem dos pulsos dos dois canais do encoder incremental a uma taxa de amostragem f_s . Definindo δ como o número de pulsos contados no contador CNT entre dois períodos de amostragem consecutivos, a velocidade angular do rotor ω_m é dada pela expressão:

$$\omega_m = 2\pi f_s \frac{\delta}{ARR} \quad (18)$$

$$\delta = CNT_k - CNT_{k-1} \quad (19)$$

Onde:

ARR : número de pulsos do encoder em uma revolução mecânica

CNT_k : valor do contador CNT no instante k

A precisão da medida da velocidade $\Delta\omega_m$ em radianos por segundo é dada por:

$$\Delta\omega_m = \frac{2\pi f_s}{ARR} \quad (20)$$

Assim, quanto maior a taxa de amostragem, menor é a precisão da medida, porém, taxas de amostragem baixas podem influenciar a dinâmica do sistema.

A série STM32F4 possui um modo específico para ser usado com encoders incrementais nos seus periféricos de timer. Nesse modo, o timer incrementa ou decrementa o valor do contador CNT em cada transição positiva ou negativa de cada um dos canais, mas apenas se os pulsos seguirem as sequências como na Figura 4. Isso permite quadruplicar

a resolução do encoder, além de possibilitar a correção de possíveis ruídos, como ilustrado na Figura 14.

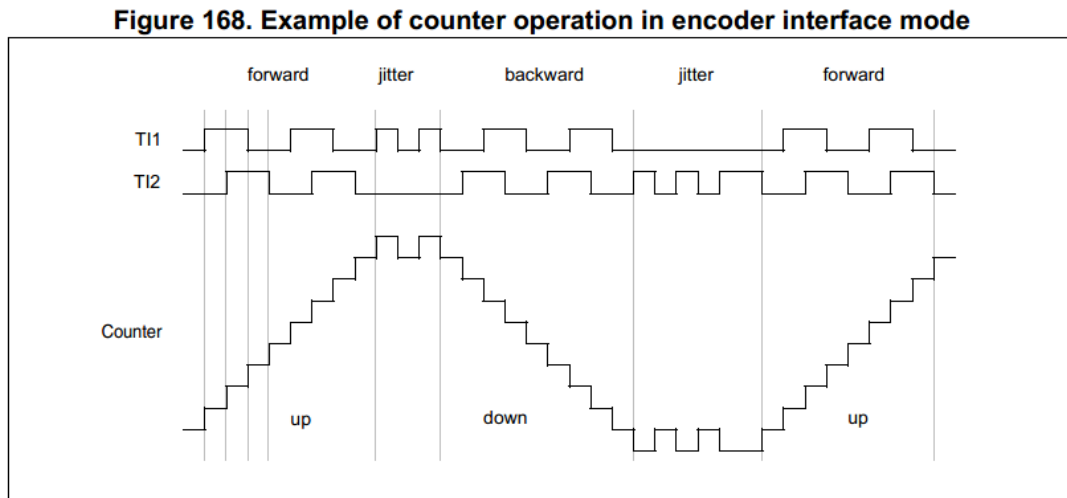


Figura 14 – Funcionamento do modo encoder dos timers da série STM32F4, extraído de (ST Microelectronics, 2014)

Assim, é possível medir velocidades positivas e negativas, que representam velocidades no sentido horário e anti-horário respectivamente. Porém, como o contador zera seu valor quando $CNT = ARR$, algumas considerações devem ser feitas para calcular corretamente a velocidade.

Primeiramente, deseja-se que a frequência de amostragem f_s seja muito maior que a frequência de rotação máxima do rotor. Nesse caso, δ terá quatro possíveis faixas de valores, como ilustrado na Figura 15.

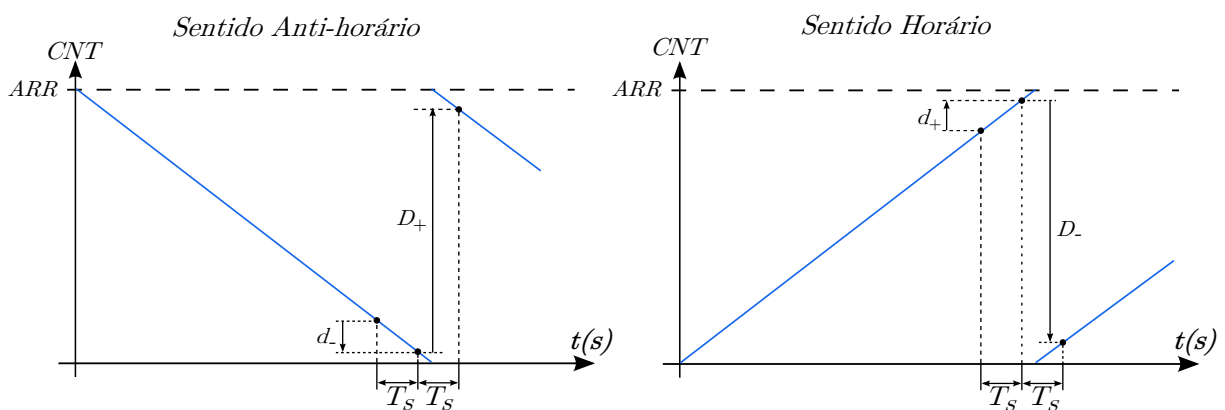


Figura 15 – Amostragem do contador do timer

Quando δ é pequeno, ou seja $\delta = d_-$ ou $\delta = d_+$, a velocidade é calculada normalmente pela Equação 18.

Porém, quando ocorre um *underflow* do contador, $\delta = D_+$, a equação a ser utilizada é:

$$\omega_m = 2\pi f_s \frac{\delta - ARR}{ARR} \quad (21)$$

Já no caso de ocorrência de um *overflow* do contador, $\delta = D_-$ e a equação agora é:

$$\omega_m = 2\pi f_s \frac{\delta + ARR}{ARR} \quad (22)$$

4.2 Motor BLDC

O motor BLDC utilizado foi o Siemens 1FT5 062 AC01, apresentado na figura 16. Esse é um motor trifásico conectado em estrela que possui sensores Hall e um encoder óptico incremental. Os principais parâmetro do motor são apresentados na Tabela 6.



Figura 16 – Motor brushless Siemens 1FT5 062 AC01

Tabela 6 – Parâmetros do motor Siemens 1FT5 062 AC01

ω_{NOM}	2000rpm
V_{NOM}	150V
I_{NOM}	3,5A
R_S	2,3 Ω
$(L_S - M_S)$	12,5mH
n_{pp}	3
K_T	0,72N.m/A
T_{elMAX}	3,6N.m
J	$4,2 \cdot 10^{-3} \text{kgm}^2$
B	$3,032 \cdot 10^{-3} \text{kgm}^2/\text{s}$

4.3 Funcionamento do programa

O programa dos dois microcontroladores é estruturado em torno da comunicação SPI, pois cada mensagem define as próximas operações de cada sistema.

O pedido de início da comunicação SPI vem do Invslove, toda vez que o mesmo termina de amostrar os conversores AD, a uma taxa de 10kHz.

Inicialmente, o mestre deve gerar o protocolo de inicialização fazendo com que $V25=1$ na primeira palavra do protocolo. Isso fará com que o escravo envie o valor de calibração $V25$ que é utilizado para converter os valores digitais de 12 bits recebidos em valores em unidades de Volts, Ampères ou graus Celsius. Em seguida, o mestre faz com que $V25=0$ e inicia sua rotina de controle.

A cada iteração da rotina de controle, o mestre usa os valores de corrente e tensão de barramento que recebeu para executar um passo do controlador PI de corrente, que fornece como saída o ciclo de trabalho δ . Em seguida, as entradas dos sensores Hall são lidas para determinar quais fases devem ser ativadas e com base nessa informação, a palavra de comando da mensagem é determinada. Por fim, a palavra de comando e o ciclo de trabalho são colocados no buffer de transmissão do SPI para serem enviadas na próxima iteração.

Paralelamente, a uma frequência de 320Hz, uma rotina de interrupção mede a velocidade do rotor como descrito no Capítulo 4.1.5. Em seguida, um passo do controle PI de velocidade é executado, fornecendo como saída a referência de corrente que é enviada ao controlador de corrente descrito acima.

Já o escravo, a cada iteração, deve gerar os sinais PWM conforme o comando recebido e em seguida amostrar novamente seus canais AD's. O escravo também realiza filtragens nas variáveis amostradas, sobretudo na corrente de barramento, que é por natureza uma variável mais ruidosa devido ao chavemaneto das fases.

4.4 Sintonia dos controladores

A sintonia inicial dos parâmetros dos controladores PI foi feita utilizando a ferramenta *rltool* do MATLAB. Essa ferramenta permite sintetizar interativamente os controladores por meio do método dos lugares das raízes.

Dentro do *rltool*, utilizou-se a opção de calcular os parâmetros dos controladores automaticamente a partir do método de Ziegler Nichols. Para o controlador de velocidade, as constantes obtidas foram: $K_{pS} = 0.0277$ e $K_{iS} = 0.111$ e para o controlador de corrente: $K_{pI} = 20.8$ e $K_{iI} = 2143$. Esses parâmetros fornecem uma saída estável, porém a resposta do sistema ainda pode ser melhorada.

A sintonia fina dos controladores foi feita utilizando uma simulação do sistema desenvolvida em linguagem C. Os parâmetros foram modificados iterativamente de forma que

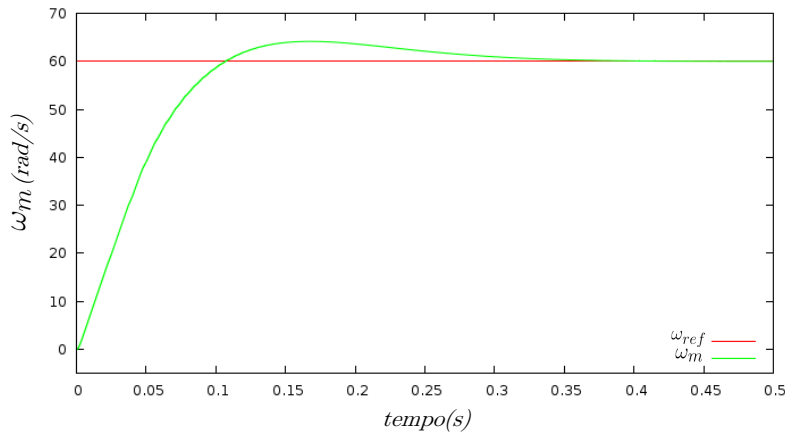


Figura 17 – Resposta simulada com $K_{pS} = 0.0277$, $K_{iS} = 0.111$, $K_{pI} = 20.8$, $K_{iI} = 2143$

se obtivessem melhores respostas de velocidade e corrente. Os novos valores obtidos da simulação foram: $K_{pS} = 8$, $K_{iS} = 2.5$, $K_{pI} = 100$ e $K_{iI} = 2500$. As respostas ao degrau simuladas com os valores obtidos em MATLAB e na simulação em C são apresentadas nas Figuras 17 e 18 respectivamente.

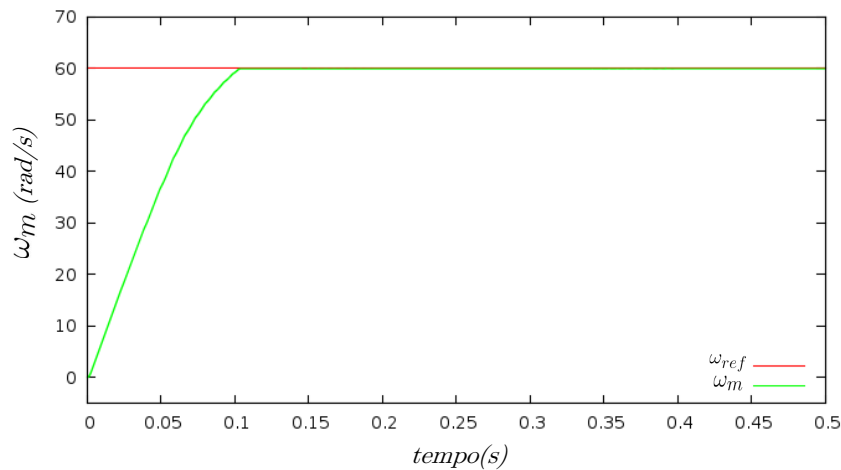


Figura 18 – Resposta simulada com $K_{pS} = 8$, $K_{iS} = 2.5$, $K_{pI} = 100$, $K_{iI} = 2500$

Em seguida, esses controladores foram testados experimentalmente. Como existem diferenças entre as grandezas físicas do sistema real e do sistema simulado, uma última etapa de sintonia dos controladores foi feita em bancada, fornecendo $K_{pS} = 0.115$, $K_{iS} = 1.85$, $K_{pI} = 10$ e $K_{iI} = 2150$.

Testes e resultados

Para a realização dos ensaios, foram utilizadas uma fonte CC Minipa MPC-303DI, um osciloscópio Agilent DSO 5014A, o motor Siemens 1FT5 062 AC 01 e a plataforma de controle desenvolvida. Uma outra ferramenta importante utilizada foi o Speedwatch do GDB, que permite extrair o valor das variáveis utilizadas nos microcontroladores. A montagem experimental é apresentada na Figura 19.

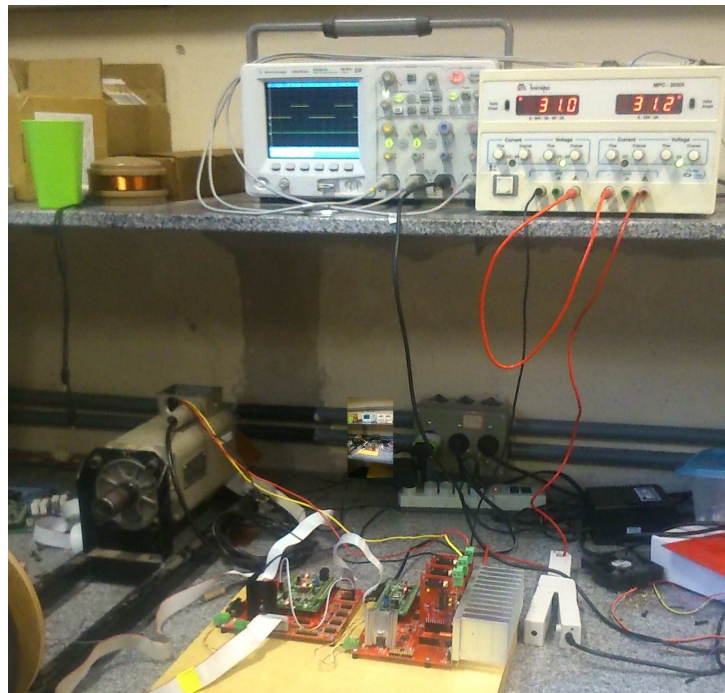


Figura 19 – Montagem experimental para a realização dos ensaios

5.1 Sincronização dos sensores Hall e fases do motor

O motor utilizado não possuía documentação que relacionasse os três sensores Hall disponíveis com as fases do motor. Para determinar essa relação, pode-se aplicar um

torque mecânico ao rotor. Essa rotação fará com que uma tensão trapezoidal seja induzida entre as fases do motor. O formato das tensões de fase V_{AB} , V_{BC} e V_{CA} e dos sinais dos sensores Hall estão relacionadas como na Figura 20

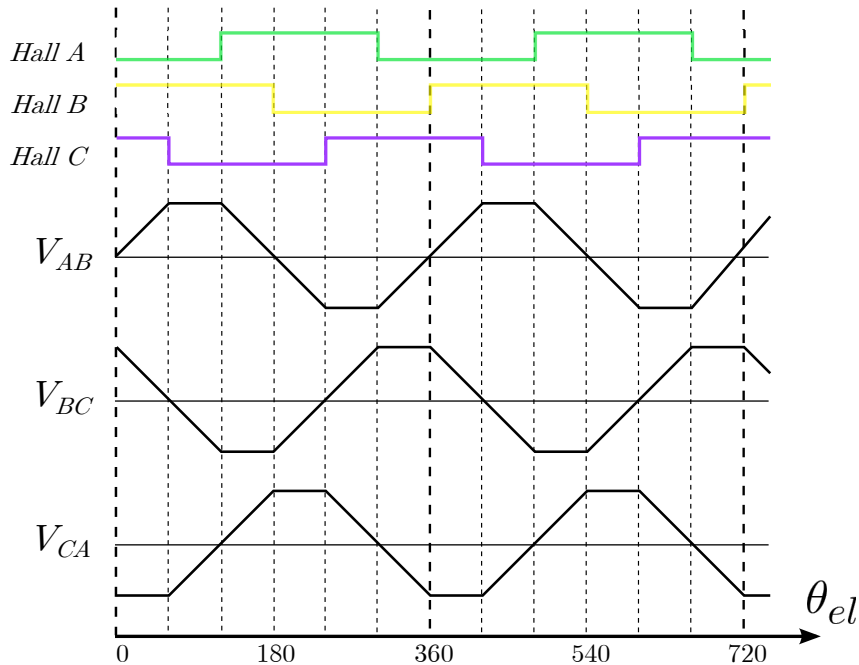


Figura 20 – Relação entre as tensões induzidas e os sensores Hall

Assim, ao observar em um osciloscópio a tensão entre duas fases quaisquer do motor e os sinais dos sensores Hall, é possível determinar o sincronismo dos sensores. Esse teste produziu o resultado da Figura 21, onde a tensão de fase medida foi V_{ab} , o que implica que o sensores em verde, roxo e rosa correspondem às fases B, A e C respectivamente.

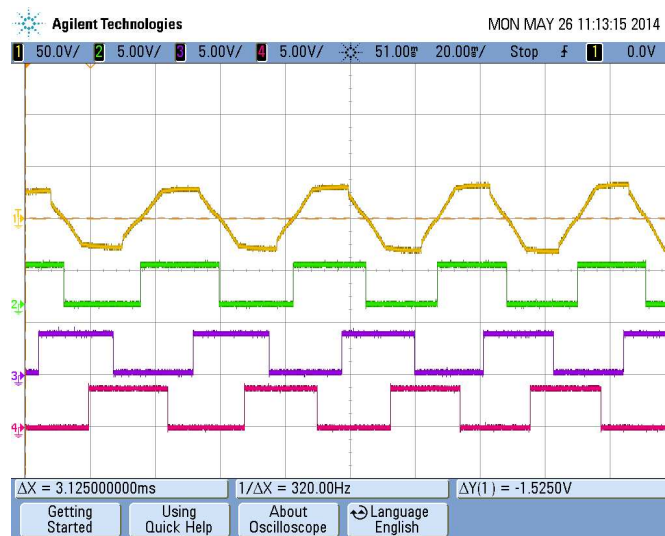


Figura 21 – Relação entre as tensões induzidas e os sensores Hall

5.2 Medição de velocidade

Como o encoder do motor utilizado nesse trabalho possui 1001 furos, a quadruplicação de resolução do modo encoder dos timers dá $ARR = 4004$.

Utilizando uma frequência de amostragem $f_s = 320Hz$, a partir da Equação 20 tem-se que a precisão da medida de velocidade será $\Delta\omega_m = 0,5rad/s$.

Para validar a medida de velocidade, o motor é acionado em malha aberta com tensão de barramento $V_{bus} = 50V$. O sinal de um sensor Hall é analisado em um osciloscópio e o período entre três transições positivas é medido, fornecendo $T_{hall} = 118,69ms$, vide Figura 22. Assim a velocidade do rotor em radianos por segundo é de $\omega_m = 2\pi T_{hall}^{-1} = 52,937rad/s$. Utilizando o *speedwatch* do GDB, foi possível extrair o valor de velocidade calculado na variável *speed*, que fornece o valor de $52,726rad/s$ como ilustrado na Figura 23.

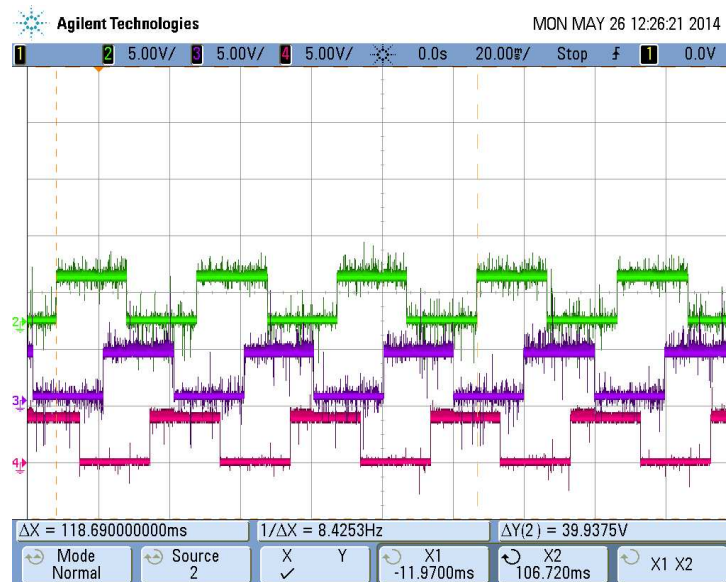


Figura 22 – Medida de T_{hall}

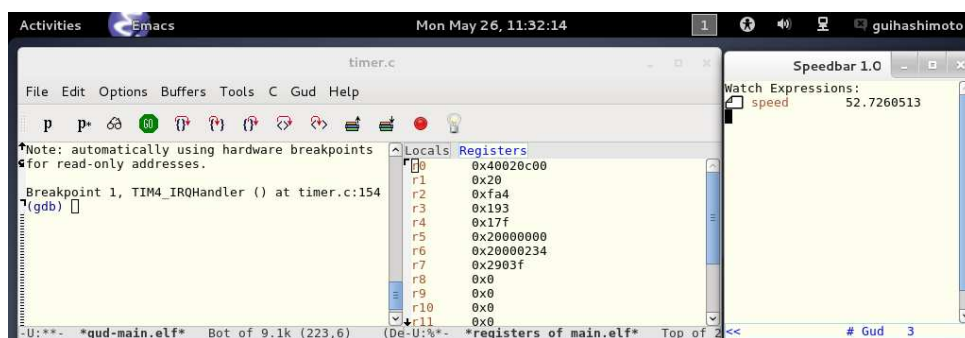


Figura 23 – Medida de velocidade obtida através do GDB

5.3 Ensaaios em malha fechada

Para os ensaios em malha fechada, os valores de velocidade, corrente e tensão do barramento são armazenados dentro do microcontrolador da Headcon e extraídos através do GDB. A Figura 24 compara a resposta ao degrau $\omega_{ref} = 60rad/s$ obtida experimentalmente com a resposta simulada em C.

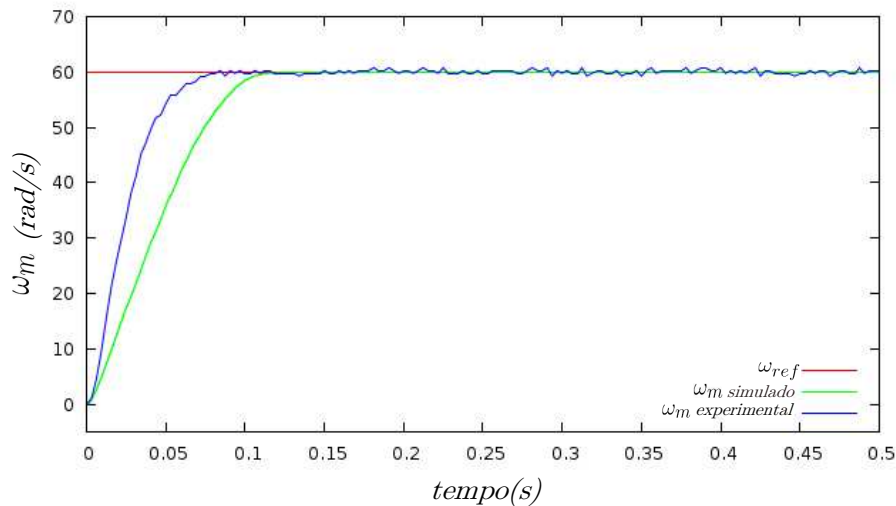


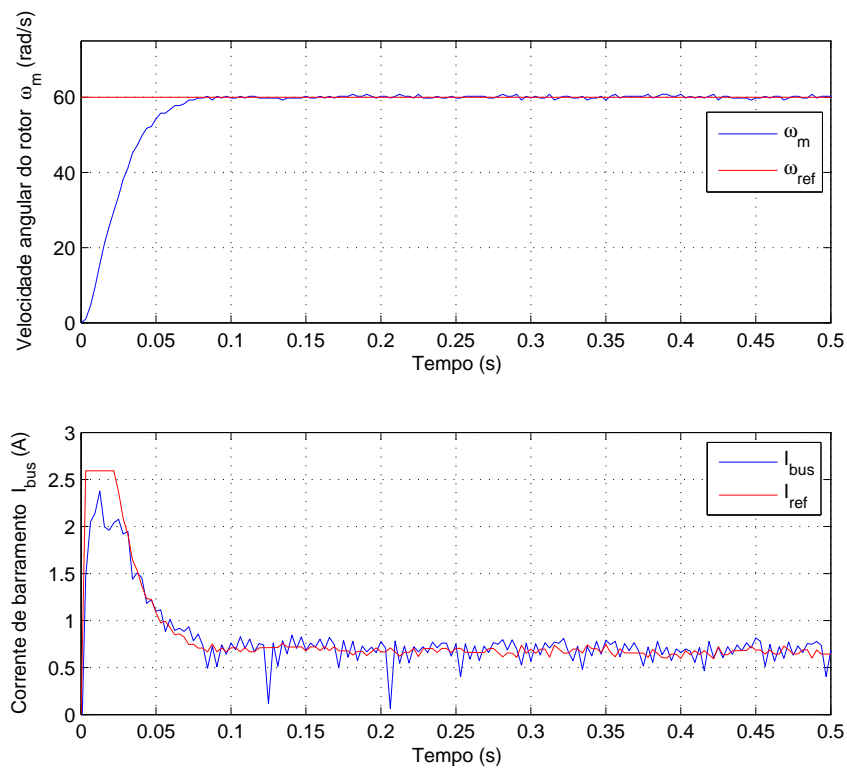
Figura 24 – Resposta ao degrau de $\omega_{ref} = 60rad/s$ experimental e simulada

Observa-se que nos dois casos, não há *overshoot*, mas a resposta do sistema real é mais rápida do que a resposta simulada. Essa diferença pode ser atribuída à diferenças nos valores do momento de inércia J e da constante de atrito viscoso B utilizadas na simulação da máquina.

As Figuras 25, 26 e 27 mostram a resposta da velocidade ω_m e da corrente de barramento I_{bus} à diferentes sinais de referência de velocidade. Em todos os casos, o controlador de corrente também teve resultado satisfatório, apesar da medida de corrente ser uma medida mais ruidosa. Também nota-se que quanto maior o degrau de velocidade aplicado, maior é a corrente de pico. Assim, para maiores alterações de velocidade, é indicado substituir o degrau por uma rampa de velocidade. Isso diminuirá a amplitude do pico de corrente.

5.4 Tempo de execução do programa

Pinos de saída foram utilizados para verificar o tempo de execução de cada iteração do programa no microcontrolador mestre, como ilustrado na Figura 28. O primeiro pino, correspondente ao sinal amarelo, sinaliza a interrupção vinda do escravo que pede o início da comunicação. O segundo pino, em verde, sinaliza o começo e o fim da comunicação

Figura 25 – Resposta da velocidade e corrente ao degrau $\omega_{ref} = 60rad/s$

SPI e o terceiro pino, em roxo, sinaliza o começo e o fim da rotina de controle de corrente. Assim, a execução de cada iteração é feita em menos de $50\mu s$.

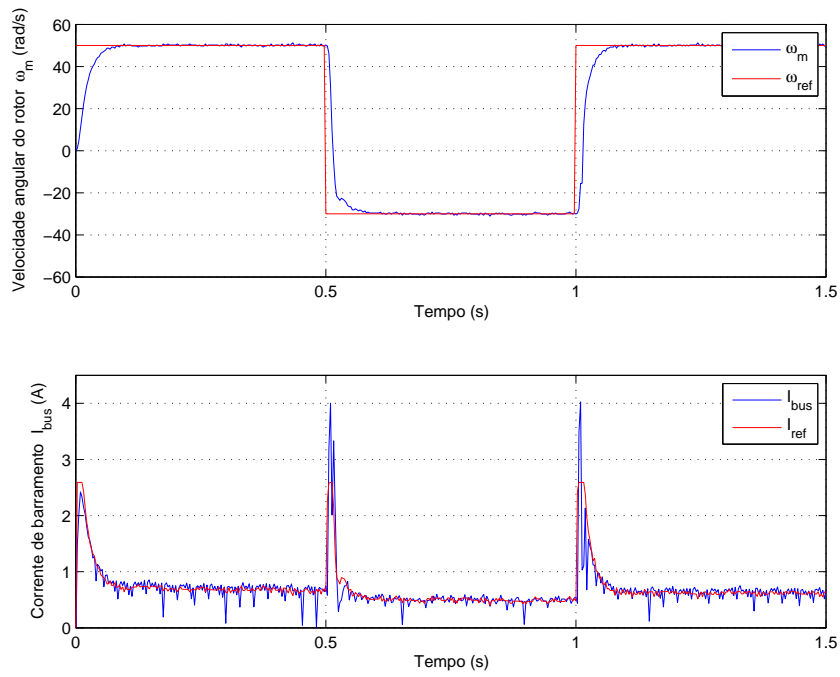


Figura 26 – Resposta da velocidade e corrente ao sinal ω_{ref}

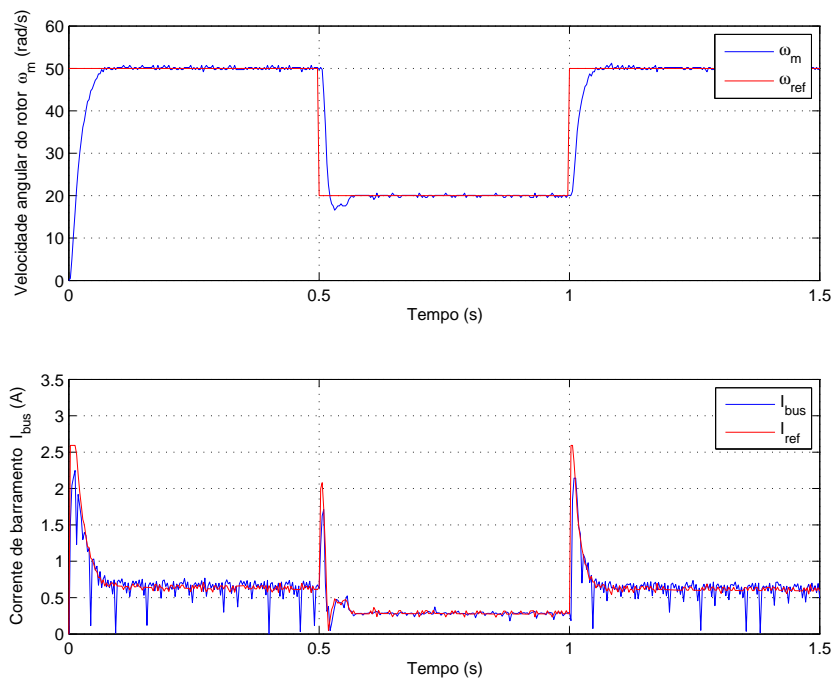


Figura 27 – Resposta da velocidade e corrente ao sinal ω_{ref}

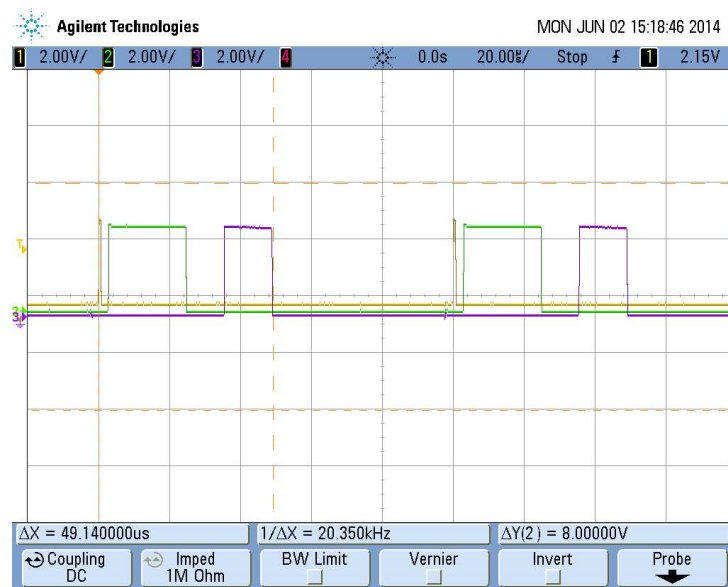


Figura 28 – Período de execução das tarefas de comunicação e controle no microcontrolador mestre.

Conclusão

O controle de velocidade de motores brushless é uma tarefa que integra conhecimentos das áreas de máquinas elétricas, controle e sistemas digitais para que possa ser realizada com sucesso. A teoria de máquinas nos permite modelar o motor para que possa ser realizado o seu controle. Esse conhecimento é fundamental para o projeto da plataforma digital de controle e sua implementação.

As ferramentas computacionais *open source* tiveram essencial importância no desenvolvimento do projeto e possuem desempenho equivalente em comparação à programas de software proprietário. Embora possuam interfaces gráficas menos desenvolvidas, um estudo mais aprofundado desses programas nos dá a habilidade de criar trechos de código e atalhos para a aplicação de interesse. Assim, o uso dessas ferramentas consegue reduzir custos do projeto sem abrir mão de desempenho e funcionalidade.

Outra ferramenta a ser destacada é a simulação do sistema em C. Como visto no capítulo 5.3, a resposta da simulação é coerente com a resposta observada experimentalmente. Embora seja possível realizar a sintonia fina do controlador em bancada, cada iteração desse processo envolve a realização de um ensaio, seguido da extração e visualização das variáveis de interesse. Já na simulação, todo esse processo é feito em poucos segundos. Além disso, a simulação permite antecipar se o sistema será instável ou se haverá picos de corrente que poderiam ser prejudiciais ao motor se o ensaio fosse realizado. Assim, o processo de sintonia fina do controlador é mais rápido e seguro quando feito em simulação.

No que diz respeito à plataforma de controle, os microcontroladores ARM M3 e M4 se mostraram muito eficientes para essa aplicação. Modos dedicados em hardware como o modo encoder dos timers na arquitetura M4 e a inserção de *dead time* nos sinais PWM na M3 substituem circuitos analógicos equivalentes que encareceriam o projeto e trechos de códigos que consumiriam tempo extra de processamento.

A topologia de um microcontrolador mestre comunicando com um microcontrolador escravo utilizando uma linha SPI fotoacoplada permite que os dois sistemas estejam eletricamente isolados, o que dá mais robustez a ambos. No entanto, a velocidade de comunicação está limitada a 3 MHz, o que diminui o tempo disponível para a realização de

cálculos no mestre. Para aumentar a velocidade de comunicação, é recomendável o uso de fotoacopladores mais rápidos e de protocolos de comunicação mais compactos.

Quanto ao controle de velocidade, observamos que os controladores PI são de fato uma solução simples e eficaz para esse problema, uma vez que os cálculos realizados levam menos de $20\mu s$.

Futuros trabalhos relacionados a esse projeto incluem controle de posição do rotor e a utilização de diferentes estratégias de controle tais como controle vetorial, *sensorless* ou fuzzy.

Referências

- ANG, K.; CHONG, G.; LI, Y. PID control system analysis, design, and technology. **Control Systems Technology, IEEE**, 2005. v. 13, n. November, p. 559–576, 2005. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1453566>.
- FITZGERALD, A. **Máquinas Elétricas: Conversão Eletromecânica da Energia, Processos, Dispositivos e Sistemas**. [S.l.]: McGraw-Hill do Brasil, LTDA., 1975.
- GAMAZO-REAL, J. C. Position and speed control of brushless dc motors using sensorless techniques and application trends. **Sensors (Basel, Switzerland)**, 2010. v. 10, n. 7, p. 6901–47, jan. 2010.
- IBRAHIM, H.; HASSAN, F.; SHOMER, A. O. Optimal PID control of a brushless DC motor using PSO and BF techniques. **Ain Shams Engineering Journal**, 2013. Faculty of Engineering, Ain Shams University, nov. 2013. ISSN 20904479.
- MONTEIRO, J. R. B. A. **Estratégias de Acionamento e Controle em Máquinas CA de Imã Permanente com Fluxo não Senoidal**. Dissertação (Mestrado) — Escola de Engenharia de São Carlos, São Carlos, Outubro 1997.
- MONTEIRO, J. R. B. A. et al. Anti-windup method for fuzzy PD+I, PI and PID controllers applied in brushless DC motor speed control. **2013 Brazilian Power Electronics Conference**, 2013. Ieee, n. 1, p. 865–871, out. 2013.
- ST Microelectronics. **RM0090 Reference Manual STM32F405xx/07xx, STM32F415xx/17xx, STM32F42xxx and STM32F43xxx advanced ARM-based 32-bit MCUs**. [S.l.], 2014. Disponível em: <www.st.com>.
- YEDAMALE, P. **Brushless DC (BLDC) Motor Fundamentals**. 2003. 1–20 p.