

# **BANCADA DIDÁTICA PARA ACIONAMENTO E CONTROLE DE MOTORES CA CONTROLADA POR DSC PROGRAMADO EM AMBIENTE MATLAB / SIMULINK**

**Newton da Silva** – newton.silva@uel.br

**Ruberlei Gaino** – rgaino@uel.br

**Marcio Roberto Covacic** – marciocovacic@uel.br

UEL – CTU – DEEL

Endereço: Universidade Estadual de Londrina – Centro de Tecnologia e Urbanismo –

Departamento de Engenharia Elétrica

Rodovia Celso Garcia Cid (PR 445), km 380

Caixa Postal 6001

CEP 86051-980 Londrina PR

**Resumo:** *As máquinas de corrente alternada são construtivamente mais simples e robustas que as máquinas de corrente contínua, possuindo também menor custo e manutenção. Entranto, quando se trata de realizar um acionamento controlado, se necessita de sistemas de controle mais sofisticados do que aqueles utilizados para as máquinas CC. A alimentação do motor é realizada através da utilização de um inversor trifásico comandado por sinais gerados por um processador. O processador produz os sinais de comando para o acionamento das chaves do inversor que alimenta a máquina. A mudança nos sinais de comando altera o sinal de saída do inversor e, conseqüentemente, a alimentação da máquina. Com a utilização do processador, é possível gerar diferentes sinais de comando para o inversor e controlar a máquina de forma distinta, bastando mudar a sua programação. Uma maneira facilitada de programação do processador é utilizar a técnica de prototipagem rápida, onde o programa é criado de modo visual através do uso de blocos e as linhas de código em linguagem C são geradas automaticamente a partir do blocos. Esta técnica, chamada de prototipagem rápida, pode ser implementada no ambiente do Matlab / Simulink e facilita o desenvolvimento do software. Neste trabalho é apresentado uma bancada didática para acionamento e controle de motores CA onde é possível desenvolver, implementar e estudar técnicas de controle e acionamento de máquinas através da programação do processador, além da realização de aulas praticas de laboratório.*

**Palavras-chave:** *Controle de Motores CA; Inversores Trifásicos; Programação DSP; Prototipagem Rápida; Bancada didática.*

## **1. INTRODUÇÃO**

A escolha pela utilização de motores CA, máquinas de indução, deve-se ao fato de esta ser mais robusta, mais barata e de fácil manutenção, se comparado com a máquina de corrente contínua, que tem como vantagem se aplicar a todas as faixas de velocidades com um controle de velocidade simplificado.

Por outro lado, a implementação de um controle mais sofisticado na máquina CA exige que a sua alimentação seja feita através de um inversor trifásico comandado por um

processador. As indústrias em geral utilizam os inversores de frequência para alimentar os seus motores CA e assim poder controlar a sua velocidade.

Ainda que seja interessante a utilização de um inversor de frequência em aulas práticas, não é possível alterar a programação do inversor para que ele responda de forma diferente do que foi programado de fábrica. Desta forma não é possível implementar uma técnica de acionamento e controle do motor além da que ele possui internamente.

O ensino de acionamento e controle de motores de CA de forma mais avançada só é possível a partir da existência de uma estrutura física composta de um inversor, comando por um processador onde sua programação pode ser alterada. Isso gera uma dificuldade para elaboração de aulas práticas mais avançadas de ensino de controle e acionamento de máquinas.

Desta forma, a implementação de uma banca didática permite ir mais além do que a utilização de um inversor de frequência comercial pois nela é possível alterar o código do programa do processador. Com isso, novas técnicas de acionamento e controle de máquinas podem ser desenvolvidas, bem como, diferentes roteiros de aulas práticas.

A alteração do código do programa do processador, ou mesmo a elaboração de novos programas é realizada em ambiente de alto nível e busca facilitar a tarefa de programação ao usuário. Utiliza-se o software Matlab / Simulink como ambiente para criação de programas.

Após o desenvolvimento do programa, é realizada a sua compilação onde o código em linguagem C é gerado pelo Simulink para o processador. Podem-se utilizar blocos próprios do simulink, ou blocos desenvolvidos pelo usuário para a geração do código em C. O Simulink funciona como um compilador que traduz todos os blocos em um programa escrito em linguagem C. Com isso, a tarefa de elaboração do programa em linguagem C para o processador é fortemente facilitada, se torna mais simples, rápida e eficiente, e pode ser realizada por um usuário não tão experiente em programação.

Esta técnica de programação de processadores e geração de códigos é chamada de prototipagem rápida (BARRIENTOS, 2011).

Neste trabalho é detalhada a implementação de uma bancada didática usada para acionamento e controle de um motor CA, sua estrutura física e a técnica de programação do processador baseada na prototipagem rápida.

## 2. DESCRIÇÃO DA ALIMENTAÇÃO DO MOTOR

A alimentação do motor trifásico é realizada através de um inversor trifásico conforme ilustra a Figura 1 (POMILIO, 2011).

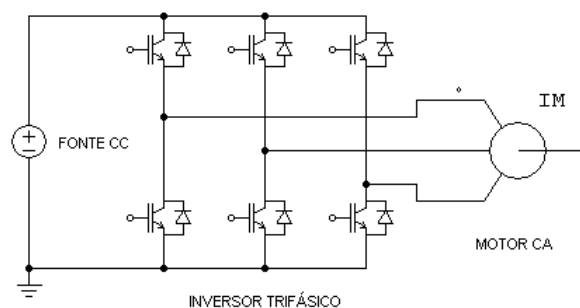
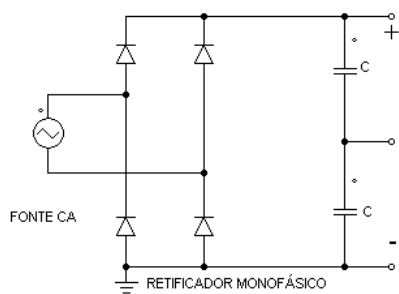


Figura 1 – Alimentação de um motor CA através de um inversor trifásico.

Uma fonte CC alimenta o inversor trifásico que é composto de seis chaves controladas, onde cada uma delas possui a ligação de um diodo em anti-paralelo. O conjunto

de duas chaves associadas em série, forma um “braço” do inversor. Cada um dos “braços” possui a conexão com os terminais de alimentação do motor, formando as saídas das fases a, b e c. O acionamento das chaves produz um sinal de tensão alternada entre as fases ab, bc e ca (tensões de linha) para a alimentação do motor. As chaves são acionadas a partir dos sinais de comando de um processador.

A fonte CC é obtida através de um circuito de retificação do sinal da rede, no caso, um retificador monofásico. A Figura 2(a) ilustra o esquema elétrico do retificador usado para conversão do sinal alternado em sinal contínuo. Na Figura 2(b) se tem a imagem do retificador implementado.



(a)



(b)

Figura 2 – Retificador monofásico: (a) Esquema elétrico, (b) Montagem.

O inversor trifásico composto de seis chaves, pode ser implementado na sua forma discreta com as respectivas chaves e seus drivers de acionamento ou, de modo integrado. Como já existe uma solução integrada, foi utilizado o circuito integrado (CI) IRAMS10UP60A (IRF 2014), fabricado pela International Rectifier. A utilização deste circuito integrado simplifica o projeto do inversor pois ele é o inversor, restando apenas fazer algumas ligações elétricas de alimentação, proteção e acionamento. A Figura 3 ilustra uma imagem deste modelo que é um módulo composto por seis IGBTs com seus respectivos drivers. Os limites de tensão e corrente de cada IGBT são de 600 V e 6 A para o modelo utilizado, valores estes, suficientes para o propósito em questão.



Figura 3 – Módulo IGBT IRAMS10UP60A integrado. Fonte (Datasheet International Rectifier).

O processador envia sinais de comando para o acionamento das chaves do inversor trifásico, que funciona como sendo um gerador trifásico de tensões e correntes alternadas que

alimentam o motor. As tensões trifásicas de alimentação do motor são sinais quadrados e uma vez que não há um filtro LC trifásico conectado na saída do inversor para filtragem da componente fundamental da tensão. Apesar disso, o motor funciona sem problemas, uma vez que os indutores do mesmo acabam atuando como filtros de modo que as correntes de linha apresentam a forma de onda de aspecto senoidal.

A Figura 4 ilustra a placa do inversor trifásico. Foram acrescentados junto ao módulo de IGBTs um circuito de proteção para o caso de ocorrer uma sobre-corrente nas chaves. Há um isolamento elétrico dos sinais de comando. Na placa há também o circuito de uma fonte simétrica necessária para alimentação dos circuitos e do próprio CI, IRAMS10UP60A. Os bornes vermelho e preto da placa são as conexões para a fonte CC e os três bornes, preto, vermelho e branco são as respectivas saídas, fases a,b e c.

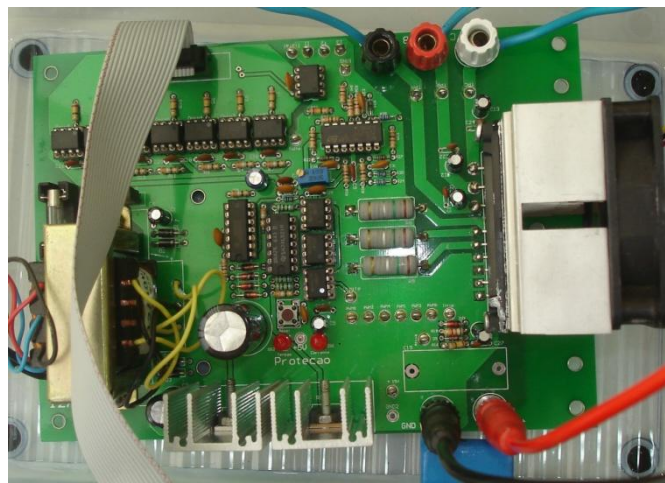


Figura 4 – Placa do Inversor trifásico.

### 3. COMANDO DO INVERSOR

O comando do acionamento das chaves do inversor é realizado através do processador da Texas Instruments modelo DSC TMS320F28335 (TEXAS INSTRUMENTS 2014) que possui internamente circuitos de geração de sinais PWM, além de outros periféricos como AD, SPI etc....

A escolha deste modelo de processador se dá ao fato de possuir todos os periféricos necessários para o controle do inversor, trabalhar com 32 bits, apresentar ponto flutuante, o que permite cálculos com números com vírgula em hardware e também possuir um clock de 150 MHz, suficiente para execução dos algoritmos de controle do motor.

Uma outra questão importante é que este modelo de processador já faz parte da biblioteca do Matlab / Simulink para programação do mesmo neste ambiente, há uma integração entre o Code Composer (programa de programação do processador) com o Simulink. Existem contrariedades de compatibilidade entre as versões do Code Composer com o Simulink. No caso em questão, foi utilizado o Code Composer V3.3 e a versão 32 bits, 2009B do Matlab / Simulink.

O processador está soldado em uma pequena placa chamada de Control Card que é encaixada em uma base (docking station), onde existem os pinos para ligações elétricas, e a comunicação com o computador. A Figura 5 ilustra uma imagem do processador. Os sinais de saída dos pinos da docking station não possuem capacidade de fornecer correntes muito intensas, desta forma, um CI é usado para acionar as chaves do inversor, atuando como driver.

O CI, além de possuir maior capacidade de corrente de saída, também apresenta uma chave que inibe todos os sinais de acionamento. Deste modo, atuando-se sobre esta chave, se inibi o funcionamento do inversor e a alimentação do motor. Isto é importante pois sempre que se deseja parar o funcionamento do processador através do computador, os pinos de saída da placa do processador ficam com níveis lógicos distintos e aleatórios, o que pode comprometer a alimentação do motor. Com a inibição dos sinais, todas as chaves do inversor ficam desligadas. A Figura 6 ilustra o circuito usado para acionamento das chaves do inversor, no caso , foi utilizado o CI SN74HCT245N.

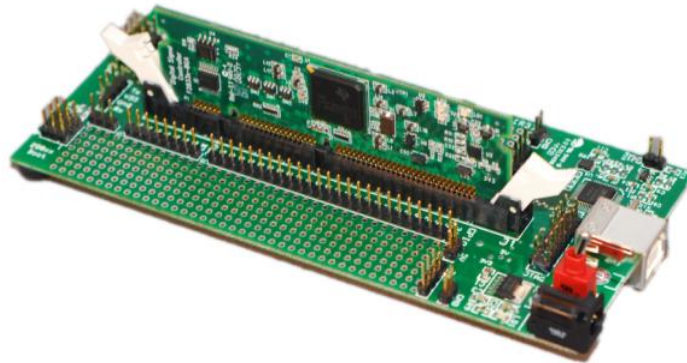


Figura 5 – Ilustração do kit DSC TMS320F28335.  
Fonte ([http://www.ti.com/graphics/tool/f28335\\_usb\\_dock.jpg](http://www.ti.com/graphics/tool/f28335_usb_dock.jpg))



Figura 6 – CI SN74HCT245N usado como driver para acionamento das chaves.

#### 4. PROGRAMAÇÃO DO PROCESSADOR

O processador é o responsável pela síntese dos sinais de comando para o acionamento da chaves do inversor. Através da programação dele, é possível implementar diferentes técnicas de controle para aplicação no motor CA.

Uma das dificuldades da utilização de um processador, é realizar a sua programação de maneira simples, rápida e eficiente de modo implementar os algoritmos de controle do motor. A elaboração dessas rotinas exige o conhecimento de programação, tanto de software como do hardware do processador em questão.

Com o avanço dos processadores, que permitem a execução de algoritmos relativamente complexos e de ambientes de programação, o desenvolvimento de programas esta se tornando uma tarefa mais simples. Este é o caso do processador TMS320F28335 que

possui uma biblioteca no ambiente do Simulink para configuração de ajustes do seu hardware permitindo assim que o processador possa ser programado neste ambiente.

Através da utilização do Simulink, a tarefa de configurar os parâmetros de hardware do processador se torna mais simples, como por exemplo, o conversor AD. No Simulink existe um bloco específico para este processador do conversor AD, onde se realizam todos os ajustes dos parâmetros do conversor AD, definindo-se taxa de amostragem, número de canais, etc.. Isto simplifica a programação do hardware do processador, uma vez que a mesma é realizada em um ambiente visual, com uma “janela” que possui abas e opções de configuração. O mesmo é válido para os demais periféricos do processador, como o caso do PWM, comunicação SPI, portas IO etc... Cada periférico possui o seu próprio bloco de programação.

Cabe ao usuário conhecer os parâmetros necessários para poder fazer os ajustes de cada periférico que será utilizado, de modo que o mesmo responda de acordo com a necessidade de quem está programando. Uma vez realizados os ajustes de hardware, se começa a elaboração do programa propriamente, que pode ser feito através da utilização de blocos do Simulink ou, criando um bloco que contem linhas de código.

Esta técnica de programação é conhecida como prototipagem rápida (BARRIENTOS 2011) e permite que usuários sem grandes conhecimentos do processador possam desenvolver programas relativamente complexos de uma forma simples, rápida e eficiente. Com isso, há uma facilidade na programação do processador e um ganho tempo no desenvolvimento do programa.

Não são todos os processadores que permitem serem programados através do Simulink já que há a necessidade da existência de uma biblioteca específica do processador para este ambiente. Alguns fabricantes disponibilizam a instalação da biblioteca do seu processador para o Simulink, quando está não faz parte do mesmo. No caso, a família C2000 da Texas Instruments faz parte da biblioteca do Simulink, na qual esta inserido o TMS320F28335. O próprio Matlab / Simulink disponibiliza um guia do usuário para programação deste processador (THE MATHWORKS 2014).

Cabe ressaltar que, o programa elaborado no Simulink precisa ser compilado para que seja gerado o código em linguagem C ao Code Composer, que é o ambiente de programação dos processadores da Texas Instruments. Existe uma comunicação entre o Simulink e o Code Composer de modo que o código é gerado automaticamente. Por fim, se faz a compilação do código no Code Composer e se envia o código final ao processador.

O trabalho do Simulink é converter todos os blocos utilizados na elaboração do programa para a linguagem C “escrita” para o Code Composer. Ele funciona como um compilador que traduz os blocos para linguagem C de forma automática.

O usuário pode alterar o código gerado pelo Simulink no ambiente do Code Composer pois ele é totalmente aberto e com comentários.

A técnica de prototipagem rápida ainda é meio recente, é uma tendência mas é necessário que os fabricantes desenvolvam bibliotecas dos seus processadores que possam ser acessadas no Simulink. O usuário não precisa ser um programador, deve apenas saber realizar a programação dos blocos do hardware do processador, além da implementação de outros blocos, ou a utilização de funções próprias do Simulink.

O usuário precisa conhecer o mínimo necessário do hardware do processador para poder realizar os seus ajustes em um ambiente visual. Uma vez ajustado o hardware, a próxima etapa é desenvolver o programa. A partir do programa construído no Simulink, gera-se o código em linguagem C que será compilado no Code Composer e transmitido para execução do processador. A alteração do código pode ser realizada tanto no ambiente do Simulink ou do Code Composer mas é mais simples de realizada no primeiro.

A utilização do TMS320F28335 permite este diferencial na sua programação, utilizar a técnica de prototipagem rápida, que expande o seu uso por pessoas de áreas que não tenham conhecimento específico de hardware de processadores mas que, através desta técnica, conseguem elaborar programas de controle digital de uma forma simplificada.

É importante salientar que toda programação do hardware do processador é realizada através dos blocos que representam o mesmo. Não é possível acessar endereços específicos de registradores do processador, nem alterar o conteúdo desses registradores. O ambiente de programação é em alto nível e não comporta o acesso ao hardware como no caso de uma programação feita através de linhas de código diretamente no Code Composer. Isto pode ser uma limitação que é contornada através de alterações no código dentro do ambiente do Code Composer. Em geral, não há a necessidade de se fazer alterações no código.

A Figura 7 ilustra um exemplo de código no ambiente do Simulink. Pode-se verificar a existência de blocos. Da esquerda para direita observam-se o bloco de ajustes do conversor AD, os blocos de programação e, no final à direita os blocos de ajustes do PWM.

No programa em questão, é realizada a leitura de dois sinais analógicos pelo conversor AD, sendo eles responsáveis pela frequência e amplitude da tensão de saída do inversor. Os sinais vão para o bloco SenCos que produz na sua saída um seno e um cosseno cuja amplitude e frequência variam de acordo com a amplitude dos sinais de entrada do AD. Assim, tem-se um gerador de seno e cosseno com controle de amplitude e frequência. Estes sinais ortogonais são transformados para os sinais a,b,c defasados 120 graus através a utilização da Transformada Inversa de Clarke. O sinais que representam as fases a,b e c servem de entrada para o bloco que realiza a modulação Space Vector, onde estes três sinais senoidais, defasados 120 graus são transformados em três sinais também defasados em 120 graus mas com “injeção” de um terceiro harmônico. Por fim os sinais de saída do bloco Space Vector são os sinais de referência para geração dos sinais PWM para comando das chaves do inversor. Há ainda um bloco de habilitação do inversor, este bloco habilita o funcionamento das chaves do inversor.

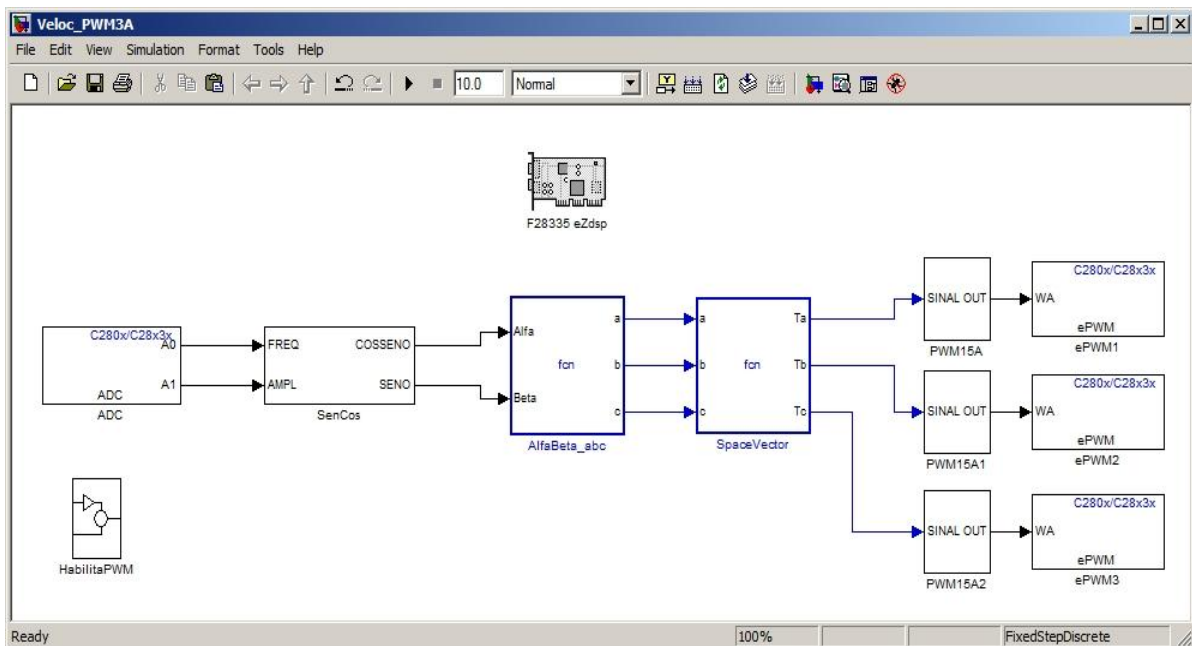


Figura 7 – Exemplo de programação no ambiente Matlab / Simulink.

De modo a facilitar a programação do processador ao usuário, foram desenvolvidos alguns blocos que executam determinada(s) função(ões), buscando-se criar uma biblioteca com os principais blocos usados no controle de máquinas CA. Isto torna o trabalho mais fácil para uma pessoa que queira elaborar um programa para controle de um motor CA. Basta utilizar os blocos que já estão prontos, de acordo com a sua necessidade. Fazendo-se a associação dos blocos, cria-se um programa de forma simples e rápida. Quanto mais blocos fizerem parte da biblioteca maior o número de possibilidades da elaboração de programas pelos usuários de forma facilitada.

Na Figura 8 é possível ver alguns blocos de hardware e também de ajustes de ganho do sinal para o AD e o PWM, bem como um bloco para conexão de um DA para visualização do sinal de saída em um osciloscópio.

A Figura 9 ilustra blocos matemáticos que foram implementados através de linhas de códigos. Pode-se visualizar os blocos para determinação do valor eficaz do sinal, valor médio, PLL monofásico e trifásico, transformações de Clarke e Park, entre outros...

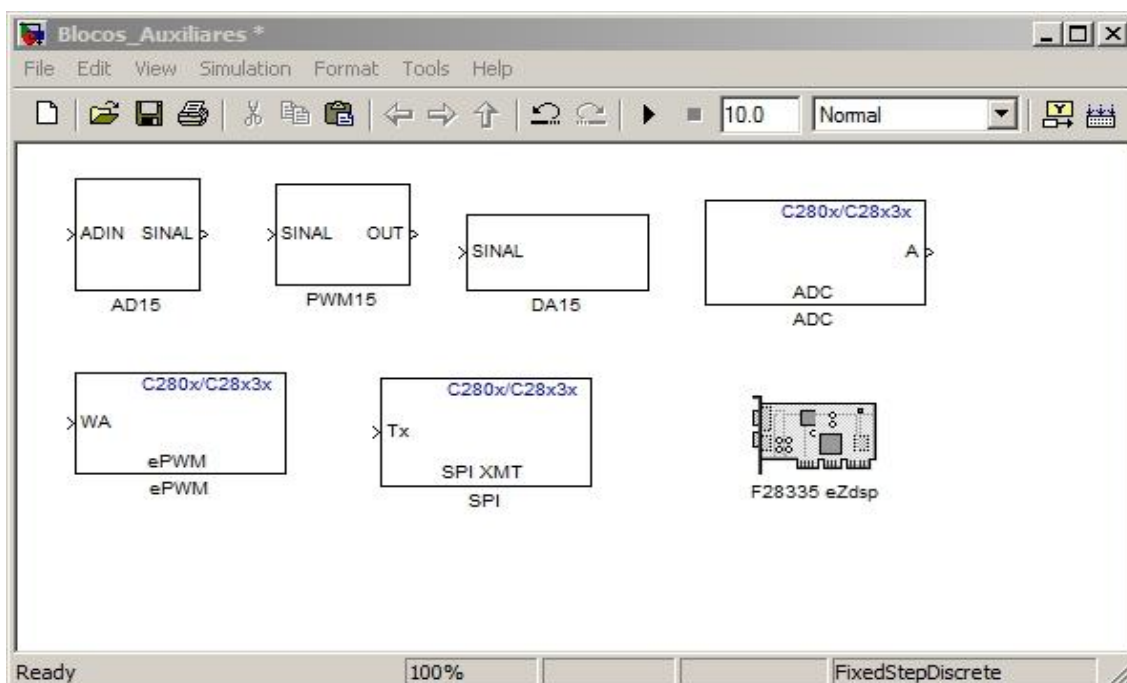


Figura 8 – Blocos para ajustes de hardware e escala: AD, PWM, DA Processador..

## 5. UTILIZAÇÃO DA BANCADA

A existência da bancada didática permite que técnicas de controle possam ser estudadas e testadas onde resultados experimentais são obtidos. Diferentes técnicas podem ser implementadas bastando realizar a alteração do software. Roteiros de aulas práticas de laboratórios podem ser elaborados, verificando-se assim o funcionamento do motor em diferentes situações de acionamento e controle.

Atualmente foi desenvolvido o controle escalar do motor CA onde é possível alterar a frequência e a amplitude da tensão de saída do inversor e verificar o funcionamento do motor. Através da alteração da frequência tem-se o controle da velocidade do motor e, com alteração da tensão, o torque. Estes controles são feitos de forma manual com dois potenciômetros. Outras técnicas estão para serem desenvolvidas com o caso do controle vetorial ou mesmo o controle escalar em malha fechada.



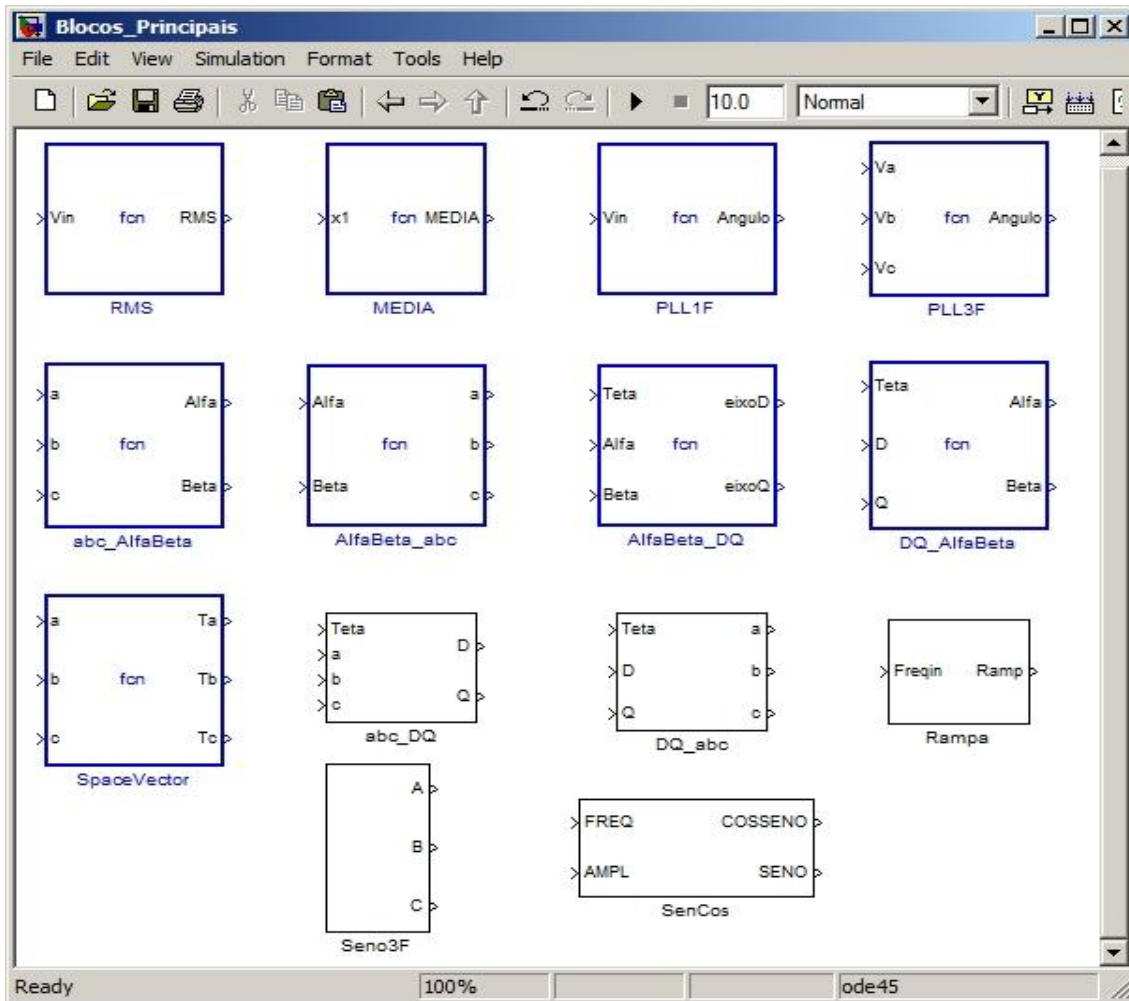


Figura 9 – Blocos matemáticos implementados.

Uma visão geral da bancada é mostrada na Figura 10 onde se pode observar a bancada completa que inclui o computador com o código do programa na tela e à esquerda a bancada com o motor.

As placas que compõem o inversor, o processador, o retificador e motor podem ser observadas na Figura 11. À direita se observa a placa de condicionamento de sinais das correntes de alimentação do motor. O controle manual da velocidade e do torque do motor é realizado através dos potenciômetros que estão no pequeno protoboard. Fuzíveis são usados para proteção da alimentação do motor, ainda que a placa do inversor já possua sua própria proteção. Mais à direita sobre a mesa branca podem ser vistas fontes de alimentação para a placa do driver e do condicionamento de sinais. As chaves contatoras não estão sendo utilizadas. Através da programação do processador podem ser testadas diferentes técnicas de controle no motor. Atualmente o controle motor está em malha aberta já que não há realimentação da sua velocidade. Algoritmos de estimativa da velocidade do motor estão sendo implementados através da aquisição dos sinais de tensão e corrente de alimentação do motor. Outros estudos estão para serem realizados.

A utilização da bancada é simples, basta “carregar” o programa no Matlab / Simulink e gerar o código. O próprio Simulink “abre” o programa Code Composer durante a geração do código. Faz-se a compilação do código gerado no Code Composer e então se transfere o programa para o processador, onde ele será executado.



Figura 10 – Visão geral da bancada didática.

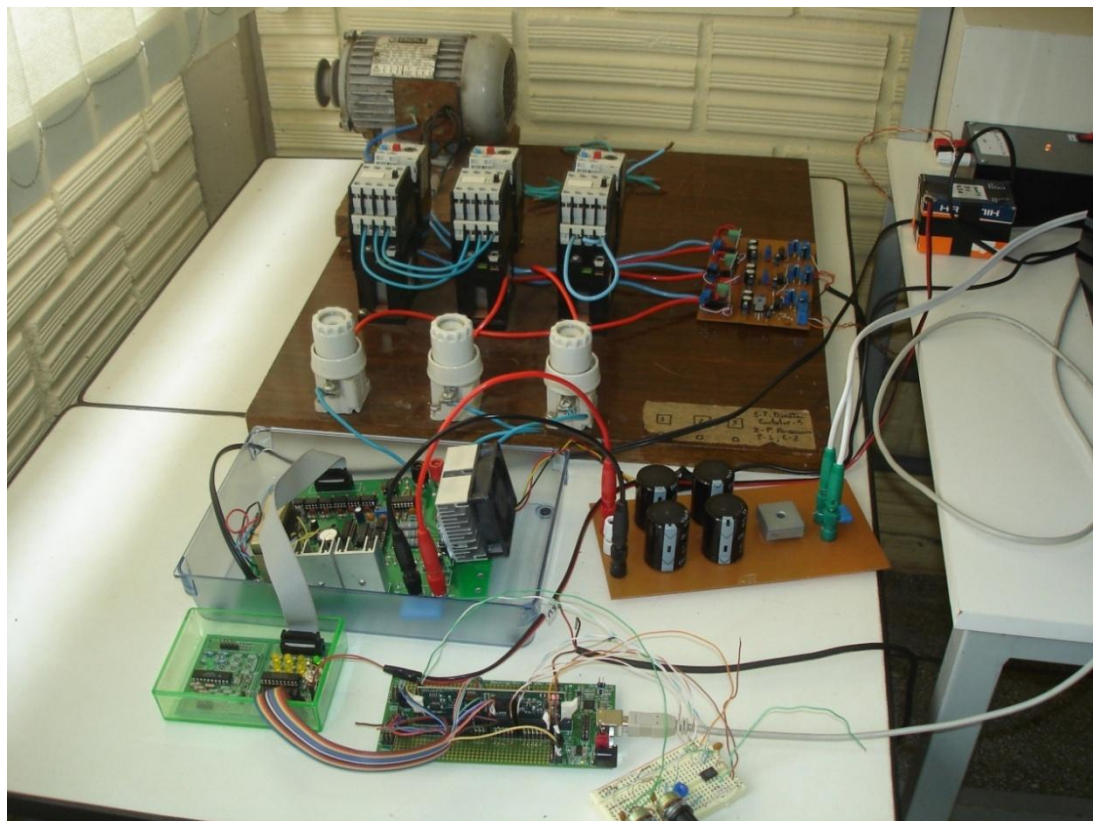


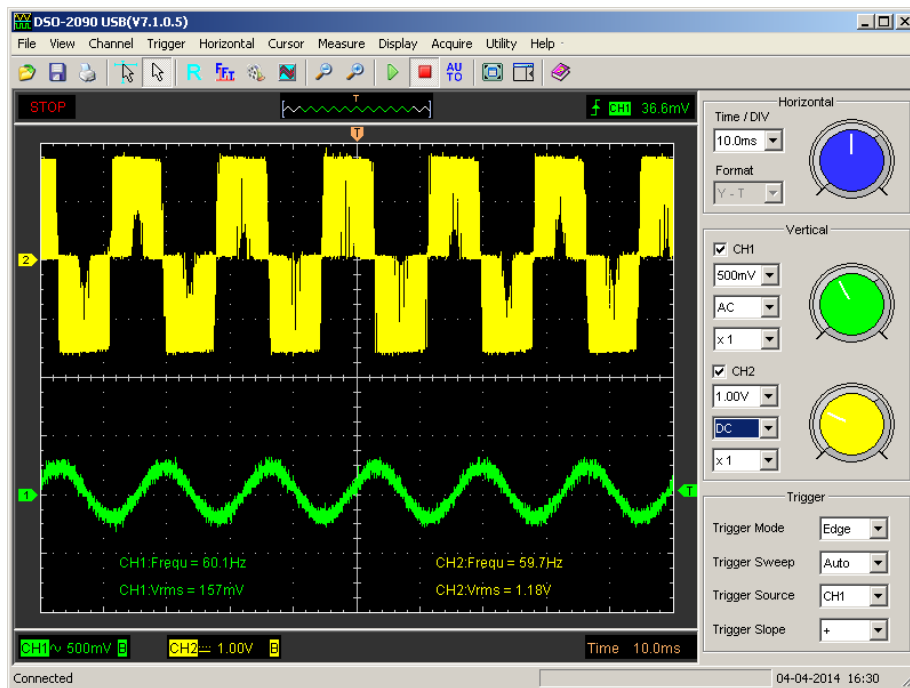
Figura 11 – Estrutura física da bancada didática: placas e motor.

Caso sejam necessárias alterações do programa, desliga-se o acionamento das chaves do inversor através da chave contida no driver. Recompila-se o novo programa no Simulink gerando-se um novo código para o Code Composer onde compila-se este código que será transferido ao processador para sua execução. Liga-se a chave do driver após o comando de execução do programa para o processador.

De acordo com a necessidade do usuário, ele pode elaborar um novo código e testá-lo na bancada, utilizando os blocos já elaborados ou criando novos blocos. O acionamento da máquina CA acaba se tornando uma tarefa de programação.

## 6. RESULTADOS EXPERIMENTAIS.

A seguir, são mostrados os sinais que representam respectivamente a tensão de linha entre as fases “a” e “b” e a corrente de linha da fase “a”. Na Figura 12 se pode observar o aspecto quadrado da tensão entre as fases e o aspecto senoidal da corrente de linha.



## 7. CONCLUSÃO

Neste trabalho foi apresentado a descrição da estrutura física de uma bancada didática usada para o desenvolvimento de técnicas mais avançadas de acionamento e controle de máquinas CA.

Além do que compõem o hardware da bancada, a questão da programação do processador também foi abordada. A técnica de prototipagem rápida mostra-se ser uma boa opção de programação, produzindo um código “aberto” e eficiente, podendo ser utilizada por pessoas que não possuem grandes conhecimentos de hardware de processadores, mas conseguem desenvolver programas relativamente complexos de uma maneira facilitada.

A programação consiste em combinar diversos blocos de modo a formar um programa maior. Cada bloco executa pequenos micro-códigos ou funções. Os blocos podem ser criados a partir de linhas de códigos ou através da associação de blocos do próprio Simulink.

A criação de um conjunto de blocos específicos para utilização em controle de máquinas facilita o desenvolvimento de futuros programas, uma vez que o programa final consiste na associação dos blocos existentes. Quanto maior o número de blocos disponíveis, maiores as possibilidades de programação.

O programa final gerado pelo Simulink apresenta um código “aberto”, podendo ser alterado no Code Composer que é o programa específico de programação dos processadores da Texas Instruments, família C2000.

Com a bancada se pode testar técnicas de controle de acionamento de máquinas CA e verificar o funcionamento de motor, bastando alterar a programação do processador. Atualmente é possível estudar a técnica do controle escalar de um motor em malha aberta.

A bancada didática usada para acionamento e controle de motores CA mostra a integração das áreas de controle de motores, eletrônica de potência, instrumentação (condicionamento de sinais para sensores de corrente e tensão) e programação de processadores digitais de sinais. Permite ao aluno ter uma visão multidisciplinar das áreas do curso de engenharia elétrica em uma única estrutura.

## 8. AGRADECIMENTOS

Os autores agradecem a FEEC, Faculdade de Engenharia Elétrica e de Computação da Unicamp onde, através do trabalho de mestrado (GARCIA, 2010) foi possível o início do estudo da técnica de prototipagem rápida pelo primeiro autor, quando estava em conclusão do seu trabalho de doutorado. O aprendizado desta metodologia e sua utilização no processador TMS320F28335 começou a partir do exemplo deste trabalho.

## 9. REFERÊNCIAS BIBLIOGRÁFICAS

BARRIENTOS, Cristian Erardo Miranda; FACULDAD DE INGENIERÍA, DEPARTAMENTO DE ELETRICIDAD, PUNTA ARENAS. Programación de DSP Através de Herramientas Disponibles en Matlab 7.0 / Simulink, 2011, Chile. Trabajo de Titulación para optar a Título de Ingeniero en Electricidad mención Electronica.

GARCIA, Fellipe Saldanha; UNIVERSIDADE ESTADUAL DE CAMPINAS, Faculdade de Engenharia Elétrica e Computação, Departamento de Sistemas e Controle de Energia. Conversores CC-CC elevadores de tensão, não isolados, com ganhos estáticos elevados, 2010. 120p. Dissertação de Mestrado.

IRF, *IRAMS10UP60A Datasheet (PD-94640 Rev I)*: Disponível em: <http://www.irf.com/product-info/datasheets/data/irams10up60a.pdf> Acesso em: 27 de março 2014

POMILIO, José Antenor. *E833 Eletrônica de Potência*. Universidade Estadual de Campinas – Faculdade de Engenharia Elétrica e Computação – Departamento de Sistemas e Controle de Energia. Campinas, SP. 2011.

TEXAS INSTRUMENTS, TMS320F28335 *Datasheet (SPRS439M June 2007 Revised August 2012)*: Disponível em <http://www.ti.com/lit/ds/symlink/tms320f28335.pdf> Acesso em 1 de abril de 2014.

THE MATHWORKS, Target Support Package 4 User’s Guide For Use with Texas Instruments C2000 (*Revised for Version 4.1 Released 2010a March 2010*): Disponível em <http://www.mathworks.com> Acesso em 1 de abril de 2014.