TIAGO DE ARRUDA RUSSOLO

Sistema de Controle de Posição Microprocessado com Servomotor CC

São Carlos
2011

TIAGO DE ARRUDA RUSSOLO

Sistema de Controle de Posição Microprocessado com Servomotor CC

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 Sistemas de Energia e Automação

ORIENTADOR: Prof. Dr. José Roberto Boffino de Almeida Monteiro

São Carlos
2011

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

R969s

Russolo, Tiago de Arruda.
Sistema de controle de posição microprocessado com servomotor CC. / Tiago de Arruda Russolo ; orientador José Roberto Boffino de Almeida Monteiro. -- São Carlos, 2011.

Monografia (Graduação em Engenharia Elétrica com ênfase em Sistemas de Energia e Automação) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2011.

1. ARM Cotex-M3. 2. Posicionamento. 3. Motor CC. I. Titulo.

FOLHA DE APROVAÇÃO

Nome: Tiago de Arruda Russolo

Título: "Sistemas de Controle de Posição Microprocessado com Servomotor CC"

Trabalho de Conclusão de Curso defendido e aprovado em 25 / 11 / 2011,

com NOTA 9,0 (nove, zero), pela comissão julgadora:

Prof. Dr. Manoel Luís de Aguiar - EESC/USP

Prof. Dr. Osmar Ogashawara - UFSCar

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

Dedicatória:

A meus pais, avós e irmãos
pelo constante apoio e orações.

Agradecimentos

Deus

Pelo seu amparo e cuidado com a minha vida.

José Roberto Boffino de Almeida Monteiro

Pelo constante apoio e pelos ensinamentos passados sempre com paciência e ânimo.

Thales E. P. Almeida, Oureste E. Batista, Geyverson T. Paula

Meus grandes amigos, pelo companheirismo e presença durante todos esses anos de estudo, além da grande ajuda em cada etapa desse trabalho.

Lissa Botezelli

Pelo seu amor, pela amizade e por suas orações diárias por mim.

Sumário

Lista de Figuras

Lista de Gráficos

Lista de Tabelas

Resumo

Esse trabalho apresenta a implementação de um sistema de controle de posição, utilizando um motor CC , e o microprocessador ARM Cortex-M3 para a aplicação em um veículo náutico autônomo. Esse sistema pode ser dividido em três partes: elementos sensores, elemento atuador e elementos de processamento. O elemento atuador é composto, basicamente, de um servomotor elétrico CC de seu conversor elétrico de potência, um *chopper* de quatro quadrantes. Os elementos de processamento são compostos por duas placas: uma com o microcontrolador e outra com função de interfacear tanto os sensores, quanto o conversor elétrico à placa de controle. A alimentação do motor é feita com modulação em largura de pulso (do inglês: PWM – pulse width modulation) gerado pelo microcontrolador que, ao captar a posição atual e a posição desejada, atuará no motor de forma a girar a direção até a posição de referência.

Palavras-chave: ARM Cortex-M3, Posicionamento, Motor CC

Abstract

This document describes the implementation of a position control system, for a DC motor, using the ARM Cortex-M3 microprocessor to use in an autonomous nautical vehicle. This system can be divided in three parts: sensors, actuator and processing. The actuator is composed by an electric DC servomotor and its power converter, which is a four quadrant chopper. The processing elements are based on two circuit boards: the first one with a microcontroller and the second one is the interface between the sensors, the power converter and the control board. The motor's input power is controlled by pulse-width-modulation (PWM) technique through the chopper. The PWM is generated by the microcontroller which calculates the difference between the desired and the actual position.

Key words: ARM Cortex-M3, Positioning, DC motor

1. Introdução

Os sistemas de posicionamento sempre foram objetos de diversos estudos com o objetivo de torná-los mais eficientes e precisos. Na indústria, o posicionamento está presente em praticamente todos os ramos, seja por meio de servomotores para usinagens precisas ou por meio de simples deslocamento de materiais entre os processos produtivos. Porém, outro tipo de aplicação com crescente interesse é o posicionamento para veículos autônomos.

Um dos exemplos mais conhecido dessa aplicação é o desafio DARPA que tem como meta o desenvolvimento de veículos terrestres autônomos para aplicação militar. Além desse desafio, existem outros para os diversos tipos de veículos, como o "Microtransat" direcionado para barcos. Porém, mesmo antes de surgirem esses desafios que incentivam pesquisas na área de controle autônomo de direção, diversos outros estudos já buscavam apresentar técnicas de controle e de mapeamento de terrenos para veículos autônomos, como, por exemplo, (SHILLER e GWO, 1991) e (PAGAC, NEBOT e DURRANT-WHITE, 1998).

É evidente que o cálculo da trajetória é um passo muito importante na navegação autônoma, porém ainda mais crucial é o sistema que põe o cálculo em prática. Desse modo, é importante que o sistema de navegação possua uma boa integração entre *hardware* e *software*, no sentido de existir segurança e confiabilidade no sistema como um todo.

Com essa visão em mente, é necessário que um bom sistema seja capaz de lidar, de maneira ágil e precisa, com diferentes dificuldades que possam aparecer. Deve ser capaz de tomar decisões próprias e de possuir um monitoramento que permita que eventuais falhas possam ser detectadas e sanadas sem que haja riscos para o equipamento e, principalmente, para as pessoas e bens externos. Um meio de aumentar a garantia de segurança está na distribuição de tarefas por diversos sistemas que trabalham independentemente, mas que são coordenados por uma unidade central. Desse modo, sem a concentração de decisões, o sistema se torna mais robusto, já que a perda de um ramo não implica na perda total do controle, além da fácil manutenção e de um desenvolvimento mais rápido.

O sistema proposto nesse trabalho é uma parte de um projeto maior que visa à navegação autônoma de um veículo náutico. Para a integração de todos os sistemas automatizados do barco, é necessária a escolha de controladores que permitam gerenciar

e interconectar os diversos sistemas pertencentes ao barco.

Nos dias atuais, é possível escolher uma série de sistemas para o controle. Porém, um dos controladores que possui boa capacidade de processamento, abrangente sistema de periféricos, versatilidade na detecção e tratamento de erros, baixo consumo, além de baixo custo, é o ARM Cortex-M3 (Yiu, 2009).

Os microprocessadores ARM tiveram início na década de 1990, quando foi fundada a Advanced RISC Machines Ltda., uma *joint-venture* da Apple, Acorn Computer Group e VLSI Technology. A partir desse ponto, em 1991, foi lançado o primeiro microprocessador, da família ARM6. Subsequentemente, outras empresas começaram a utilizar a arquitetura ARM, levando-a para diversos equipamentos como telefones celulares, PDA's e outros que necessitavam de processamento embarcado. Posteriormente, diversas outras famílias e arquiteturas foram projetadas, com uma evolução que passa pela família ARM 7TDMI, ARM11 e, até os dias atuais, pela a família Cortex, divididas em aplicações voltadas a sistemas operacionais (Cortex-A), aplicações de tempo real que necessitam de altíssimo tempo de resposta (Cortex-R) e aplicações para sistemas embarcados e sistemas de controle, com alta eficiência (Cortex-M) (Yiu, 2009). A Figura 1 ilustra a evolução da arquitetura ARM, descrita acima:

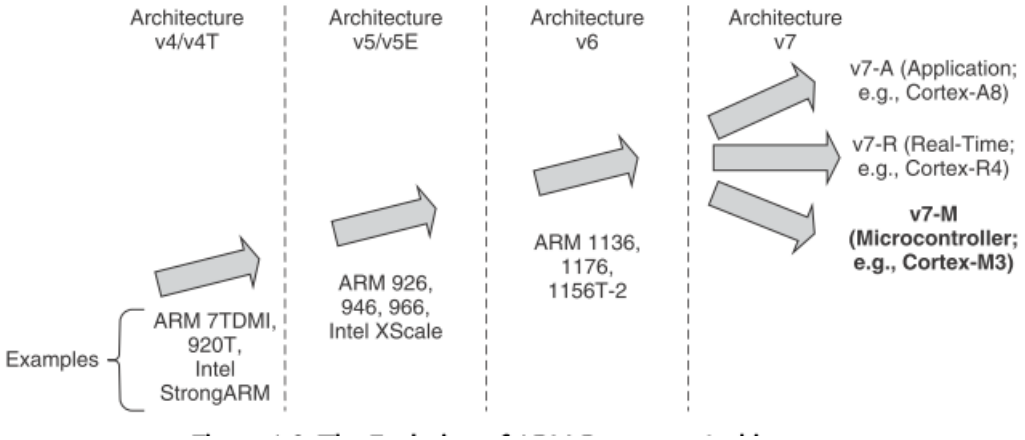


Figura 1 - Evolução da arquitetura ARM (Yiu, 2009)

Com uma quantidade significativa de desenvolvedores, é possível encontrar diversas ferramentas para a programação desses microprocessadores. Um exemplo dessa versatilidade é a existência de ferramentas de domínio público que permitem a programação, compilação de código e realização de diagnóstico, sem a necessidade de se comprar ou pagar pela sua utilização. No presente trabalho, foram utilizadas, para a interface com o microprocessador, ferramentas completamente livres de custo, mostrando assim a possibilidade de que qualquer pessoa, com o conhecimento adequado, consegue

desenvolver aplicações utilizando o microprocessador ARM.

Para acompanhar toda a tecnologia empregada no controle, é necessário que se tenha um sistema de potência semelhantemente idealizado. Para que isso seja possível, é necessário verificar as reais necessidades do projeto para que, de um modo geral, não existam excessos, tornando a aplicação complicada e menos objetiva. Com essa abordagem em mente, o sistema estará menos sujeito a eventuais falhas, tendo um diagnóstico mais simples no caso de erros, facilitando na integração com os diversos sistemas que o cerca.

No trabalho, serão abordados aspectos do projeto e da construção do sistema de posição, partindo do levantamento de dados do motor e passando pelo seu modelo em função de transferência que foi inserido no *software Matlab*®. A seguir, será demonstrado o projeto do sistema de potência para o motor, o sistema de controle que utilizará o ARM Cortex-M3, além da sua utilização e programação, e, por fim, a integração do projeto como um todo.

2. Materiais e Métodos

2.1 O motor de corrente contínua

O motor de corrente contínua (CC) foi o motor escolhido para realizar a aplicação de posicionamento. Esse tipo de motor possui, como característica construtiva, duas estruturas magnéticas básicas: o estator e o rotor.

O estator de um motor CC pode ser constituído de um enrolamento de campo ou de imãs permanentes. Em ambos os casos, o objetivo do estator é fornecer ao rotor o campo magnético que fará com que o rotor se mova quando aplicada uma corrente elétrica. O circuito do rotor é denominado por circuito de armadura

Há também diferentes modos de excitação das máquinas de corrente contínua. Um desses modos é o da excitação independente (Figura 2), para o qual são necessárias duas fontes de energia. Uma delas deve alimentar o rotor e a outra é responsável por criar o campo magnético do estator. Um segundo tipo de excitação é o paralelo ou "*shunt*" (Figura 3). Nesse caso, é necessária apenas uma fonte que alimentará tanto a bobina do rotor como a do estator

Outro modo de excitação é feito com a utilização de imãs permanentes no estator do motor de corrente contínua, sendo que a única alimentação necessária para que o motor gire é a do rotor. No trabalho em questão, esse é o tipo de motor que será estudado a fim de se realizar o controle de posição.

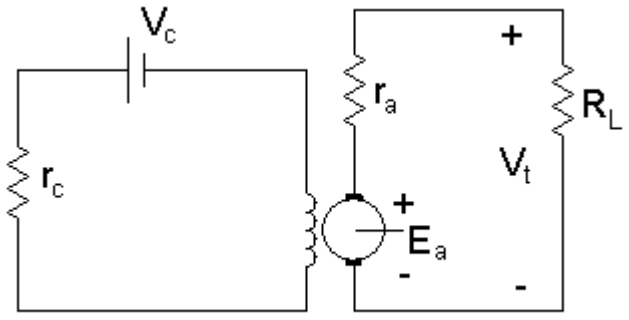


Figura 2 - Motor CC – Excitação Independente

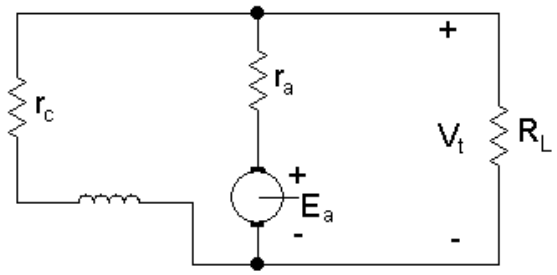


Figura 3 - Motor CC - Excitação Paralela

O rotor é um componente que possui um núcleo ferroso com enrolamentos em sua superfície. Esses enrolamentos formam as bobinas nas quais serão aplicadas correntes elétricas que, por sua vez, resultarão na interação entre essa estrutura (armadura) e o estator (campo), ocasionando o aparecimento de um fluxo magnético responsável pelo movimento do motor.

Quando aplicada a corrente elétrica ao rotor, surge um campo magnético que induz uma força com o objetivo de alinhar o campo magnético gerado na bobina do rotor com o campo magnético do estator, ocasionando assim, o movimento do motor. Ao se movimentar o rotor, outra estrutura existente no motor, chamada de comutador, é responsável por desenergizar uma bobina e energizar a seguinte, causando um novo desequilíbrio nas forças magnéticas, resultando em um novo impulso no rotor. Portanto, conforme o movimento é realizado, o motor tende a continuar em movimento, desde que seja suprida uma diferença de potencial entre seus terminais do motor, suficiente para que a força exercida pela variação do fluxo magnético seja maior que as forças que seguram o motor parado.

Uma maneira mais adequada de se explicar o funcionamento do motor pode ser feita por meio da análise da interação entre campos magnéticos do rotor (armadura) e o estator.

Quando se injeta uma corrente através de um dos fios do rotor, este sofre a ação de um campo magnético com força F demonstrada na equação (1), onde “B” é a intensidade do fluxo magnético, “l” é o comprimento do fio e “i”, a corrente. Quando a espira do rotor começa a girar, seu torque começa a aumentar pelo aumento do fluxo “ϕ” (2) até um máximo e então diminui até o ponto em que o torque seria zero. Nesse momento, em um motor de corrente contínua comum, ocorre a atuação do comutador e, então, o movimento inicia-se novamente e assim sucessivamente de modo a aumentar a velocidade de rotação do motor até o equilíbrio (FITZGERALD, JUNIOR e UMANS, 2006) (OLIVEIRA, AGUIAR e VARGAS, 2005).

$$F = B \times l \times i \qquad (1)$$

$$T = K \times \phi \times I_a \qquad (2)$$

Onde:

- F – Força exercida pelo campo magnético sobre a espira;
- B – Intensidade de fluxo magnético;
- l – Comprimento da espira;
- i – Corrente elétrica que circula na espira;
- T – Torque elétrico resultante;
- K – Constante de torque elétrico;
- ϕ - Fluxo magnético por uma bobina;
- I_a – Corrente de armadura.

2.2 Abordagem do motor CC por função de transferência

Por meio da equação 1, é possível verificar que o torque é diretamente proporcional à corrente aplicada ao motor. Isso demonstra a possibilidade de se controlar o torque gerado pelo motor através do controle da corrente. Porém, o objetivo da aplicação é o controle da posição. Desse modo, para se obter uma representação de fácil implementação no controle digital, uma abordagem por função de transferência é necessária. Para isso, um modelo semelhante aos apresentados na Figura 2 e Figura 3 de motor CC é utilizado, levando-se em consideração o modelo de armadura demonstrado na Figura 4 (MESSNER e TILBURY, 1998).

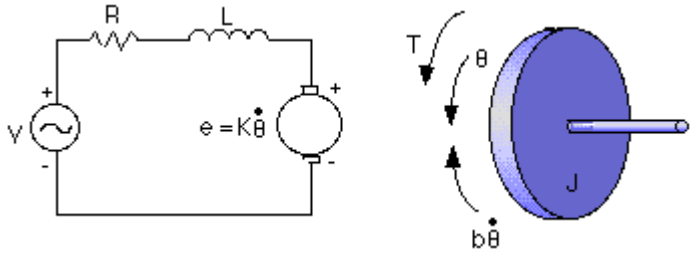


Figura 4 - Modelo de motor para equacionamento por espaço de estados (MESSNER e TILBURY, 1998)

O significado de cada um dos símbolos utilizados na Figura 4 pode ser visto na Tabela 1.

Tabela 1 - Símbolos utilizados para descrição do motor CC

J	Momento de Inércia do motor
b	Constante de amortecimento do sistema mecânico
θ	Posição do rotor
$\dot{\theta}$	Velocidade do rotor
$\ddot{\theta}$	Aceleração do rotor
K	Constante do motor ($K=K_t=K_e$)
R	Resistência da armadura
L	Indutância da armadura
V	Tensão aplicada ao motor

Portanto, é necessário obter alguns desses dados a fim de se modelar o motor de forma adequada. Porém, antes da demonstração de um método de obter essas constantes, deve-se analisar o sistema como um todo (OGATA, 2003).

A partir do circuito elétrico da Figura 4, é possível deduzir as equações (3), (4) e (5), demonstradas abaixo.

$$V - K_e\,\dot{\theta} = RI + L\frac{dI}{dt} \qquad (3)$$

$$J\ddot{\theta} + b\dot{\theta} + F = T \qquad (4)$$

$$T = K_t I \qquad (5)$$

Continuando a análise das equações, sabe-se que K_e tem a mesma magnitude que K_t. Portanto, essas duas variáveis serão substituídas nas equações (3) e (5). Além disso, é possível substituir o torque da equação (5) na equação (4). Portanto, temos o seguinte resultado, como demonstrado pelas equações (6) e (7):

$$V = K\,\dot{\theta} + RI + L\frac{dI}{dt} \qquad (6)$$

$$J\ddot{\theta} + b\dot{\theta} = KI \qquad (7)$$

Para o próximo passo é necessário aplicar a transformada de Laplace a fim de passar as equações (6) e (7) do motor para o domínio da frequência, com o intuito de representar o motor em função de transferência, sabendo que o equacionamento do motor, apresentado acima, foi escolhido de modo a se obter um modelo de 2ª ordem (OLIVEIRA, AGUIAR e VARGAS, 2005).

As equações (8) e (9), mostradas abaixo, são usadas para se obter a representação matemática em forma de função de transferência para a elaboração de um diagrama de blocos que ilustrará o motor.

$$V(s) = (L\,s + R)\,I(s) + K\,s\,\theta(s) \qquad (8)$$

$$K \times I(s) = s\,(J\,s + b)\,\theta(s) \qquad (9)$$

Utilizando o *software Matlab*®, e o equacionamento demonstrado acima, é possível elaborar o diagrama de blocos do motor, mostrado, a seguir, na Figura 5:

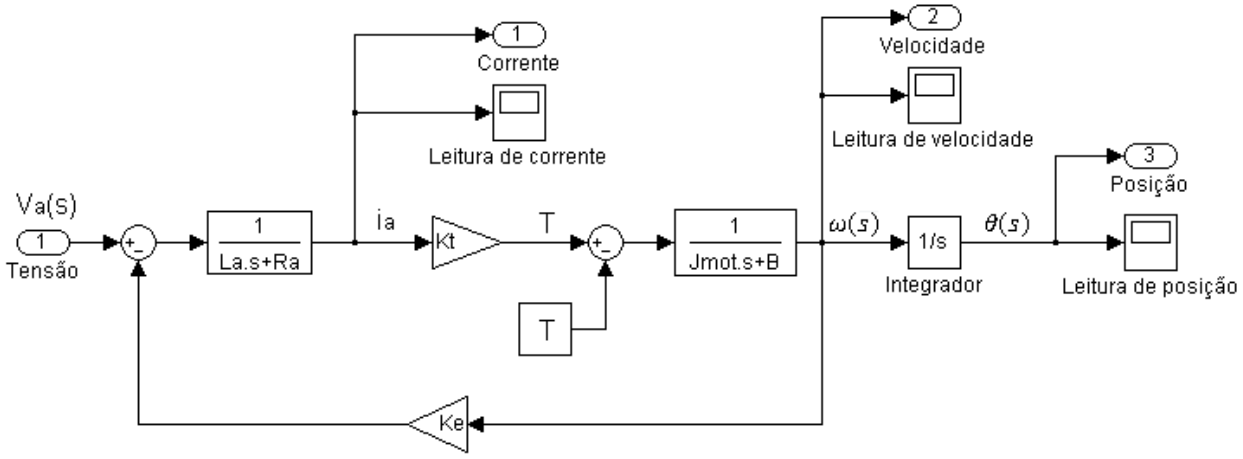


Figura 5 - Diagrama de blocos do motor CC

A vantagem de se obter um modelo que represente de forma satisfatória o comportamento do motor pode economizar um bom tempo no decorrer do projeto quando for necessário passar todo o controle para o microcontrolador, responsável por controlar a planta real. Além disso, com a integração entre esse modelo e o software *Matlab*®, existem várias outras ferramentas de análise e cálculos que podem ser integradas, de

modo a se conseguir capturar diversos dados, como a simulação do controle, empregando blocos PID.

2.3 Determinação dos parâmetros do motor CC

No tópico anterior, foi feita a abordagem teórica para a construção do modelo do motor de corrente contínua. Prosseguimento com a modelagem é preciso realizar o levantamento de dados relativos ao motor que se deseja simular. Para isso, como a base dos cálculos foi feita referente à Figura 4, os dados necessários podem ser todos vistos nessa mesma figura e na Tabela 1.

Com o intuito de se organizar a aquisição e determinação dos parâmetros do motor, foi utilizado o método proposto em (OLIVEIRA, AGUIAR e VARGAS, 2005).

2.3.1 Determinação da resistência de armadura

Primeiramente, é necessário obter a resistência elétrica do rotor. Essa medida é feita com a utilização de um ohmímetro conectado aos terminais da armadura do motor de corrente contínua. Um detalhe que deve ser observado é a variação da resistência do motor com relação à posição do rotor. Desse modo, como proposto na referência, são realizadas algumas medidas e, na posição com menor valor encontrado, mantém-se o rotor e realizam-se as outras medidas referentes a essa posição.

2.3.2 Determinação da indutância de armadura

O segundo passo é medir a indutância da armadura do motor CC. Nesse caso, se estiver disponível, um medidor de indutância, conectado aos terminais da armadura do motor, pode ser utilizado. Esse método tem a vantagem de determinar rapidamente o valor da indutância, sem a necessidade de nenhum outro equipamento, como osciloscópio e fonte.

2.3.3 Determinação da constante K_e

Para o cálculo da constante K_e do motor, utiliza-se como base a equação 6 referente aos termos elétricos do motor. Quando o motor entra em regime permanente, pode-se dizer que $\frac{dI}{dt} = 0$. Portanto, com o termo $L\frac{dI}{dt}$ anulado, o resultado é demonstrado na equação 10.

$$K_e = \frac{V(t) - R.i(t)}{\omega(t)} \qquad (10)$$

Desse modo, medindo a tensão de entrada e sabendo o valor da resistência, da corrente medida com um amperímetro e da velocidade, mensurada com um tacômetro, obtém-se a constante K_e.

2.3.4 Determinação dos coeficientes de atrito estático (b) e viscoso (F)

Para se determinar os coeficientes de atrito estático e viscoso, é necessário realizar algumas suposições, referentes ao regime permanente. Nesse caso, pode-se dizer que $\frac{d\omega}{dt}$=0 e $\frac{di}{dt}$=0. Assim, do ponto de vista energético, é possível dizer que a potência entregue pela fonte para o motor está sendo usada para vencer as perdas de movimento e as perdas ôhmicas da armadura. Desse modo, pode-se reescrever a equação 4 na equação 11.

$$B\,\omega + F = K_t\, I_a = T_R \quad (11)$$

E, do balanço energético, adicionando-se o torque T_R, tem-se a igualdade mostrada na equação 12, que demonstra a potência total entregue ($V_a.I_a$) como sendo a soma das perdas ôhmicas ($R_a.I_a^2$) e das perdas mecânicas, proporcionais ao aumento da velocidade ($T_R.\omega$).

$$V_a\, I_a = R_a\, I_a^2 + T_R\, \omega \quad (12)$$

Reorganizando a equação 12, é possível isolar o termo relativo às perdas mecânicas (equação 13) e, a partir de uma curva T_R x ω, calcular graficamente os valores de B e F, como mostrado na Figura 6:

$$T_R = \left[\frac{V_a - R_a I_a}{\omega}\right] I_a \quad (13)$$

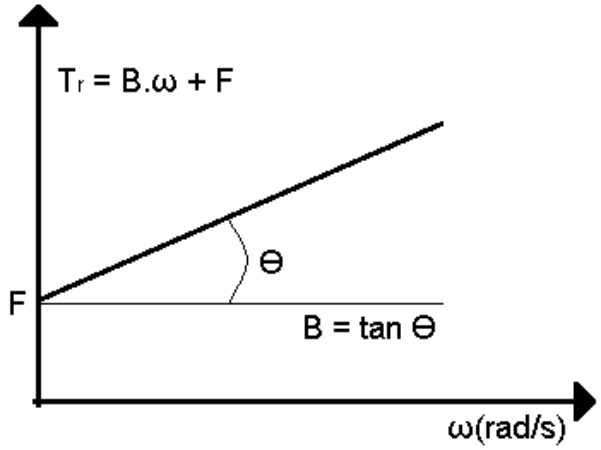


Figura 6 - Gráfico para determinação de B e F

2.3.5 Determinação do momento de inércia (J)

Um dos meios de se determinar o momento de inércia do motor é através da sua constante de tempo. Para a obtenção dessa constante de tempo, é necessário partir o motor e deixá-lo atingir o regime permanente. Atingindo o regime, é preciso monitorar a velocidade do motor, desligar a sua fonte de alimentação, aguardar a sua parada e identificar, em uma curva ω x t do motor, o tempo "t_b" tal que ω' = 0.386 ω_O, como demonstrado na Figura 7. Por fim, com esses dados em mãos, é possível calcular J a partir da equação 14.

$$J = B\ t_b \qquad (14)$$

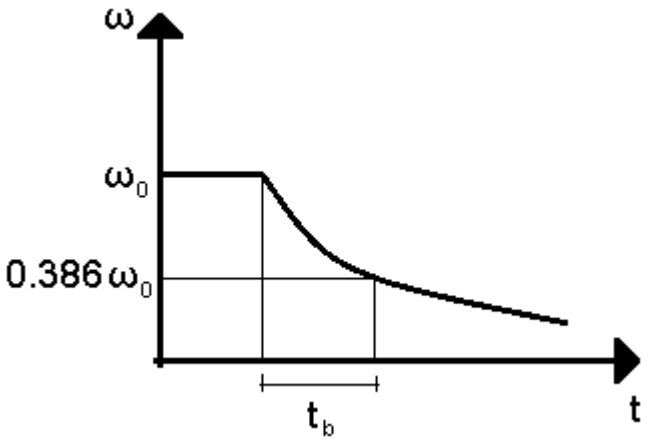


Figura 7 - Constante de tempo mecânica

2.3.6 Determinação da constante Kt

Para se determinar a constante Kt, é possível aplicar um método baseado no atrito viscoso e na corrente de armadura, porém, numericamente, Kt possui o mesmo valor de Ke. Dessa forma, como Ke já foi determinada, sabe-se também o valor de Kt.

2.4 Ações de controle envolvidas

A partir de um modelo que retrata o motor físico, o próximo objetivo será preparar simulações com o propósito de identificar os requisitos necessários na realização do controle de posição do motor de corrente contínua.

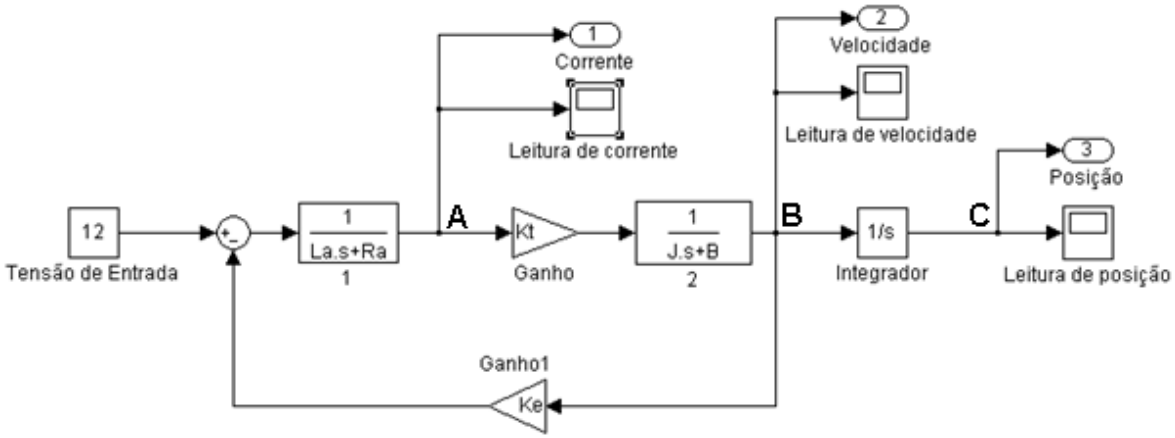


Figura 8 - Aplicação de tensão ao modelo do motor de corrente contínua

Primeiramente, verifica-se do comportamento do motor, modelado no *Simulink*®, em malha aberta. Essa verificação é importante para avaliar se o modelo está condizente com o motor real, do ponto de vista da corrente e da velocidade. Para isso, deve-se utilizar o bloco do motor construído anteriormente aplicar uma tensão de entrada e observar as saídas de corrente e velocidade do motor.

Quando se diz que o controle está em malha aberta, significa que entre a saída e a entrada do sistema não existe nenhuma informação propagada. Em outras palavras, pode-se dizer que a entrada do sistema não se regula para que a saída seja como desejado. Tomando como exemplo o modelo do motor mostrado na Figura 8, não é possível regular a posição do sistema. Isso ocorre pois a entrada do sistema, que seria posição, não tem a informação de como está a saída.

A Figura 8 mostra o modelo criado no *Simulink*® para visualização da resposta do motor à uma tensão de 12V aplicada nos terminais de entrada. Neste sistema foram adicionados três leituras. A primeira é relativa à corrente, a segunda é relativa à velocidade e, na terceira, é mostrada a posição do eixo do motor. Uma grande vantagem desse *software* é a possibilidade de se verificar, de um modo gráfico, o comportamento do motor, imediatamente ao final da simulação.

Após a simulação do motor em malha aberta ter sido realizada, o próximo passo será o controle da posição. Para isso, serão feitas três realimentações; corrente, velocidade e posição.

A realimentação é a técnica empregada para se controlar uma variável de entrada desejada, de acordo com a saída. Isso basicamente significa que um sinal de saída, seja ele corrente, velocidade ou posição, atua na entrada juntamente com a referência.

No caso do motor, ainda na Figura 8, os pontos A, B e C são referentes aos locais onde os sinais de saída serão aquisitados. No ponto A, tem-se o comportamento da corrente do motor, no ponto B está a informação da velocidade e, no ponto C, a posição do eixo do motor. Como o controle deverá atuar sobre estas três variáveis, será utilizado um controle em cascata. Para isso, cada malha de controle será ajustada separadamente, do laço mais interno (o da corrente) para o laço mais externo (o da posição).

O tipo de controle que será empregado nas malhas de corrente, velocidade e posição é o Proporcional-Integral-Derivativo, ou PID. Esse tipo de controle é de simples aplicação e também de fácil implementação digital. Além disso, é um controlador utilizado há bastante tempo pela indústria em diversas aplicações como controle de máquinas ferramenta, bombas e motores elétricos. Sua função de transferência pode ser descrita, no domínio da frequência, de acordo com a equação 15.

$$G_{PID}(s) = Kp + \frac{Ki}{s} + Kd\ s \qquad (15)$$

Como não está no escopo deste trabalho buscar o ponto ótimo do controle do motor, mas sim aplicar conhecimentos de diversas áreas para conseguir controlar a posição, o algoritmo de auto-ajuste de PID do *Simulink*® é uma ferramenta prática de análise da resposta do motor. O funcionamento desse algoritmo é baseado na resposta ao degrau do sistema. Assim, é feita a linearização da planta com base nos blocos de controle, sendo possível a escolha, em uma barra, da velocidade desejada da resposta, como mostrado na Figura 9. Para isso, o *software* calcula valores de ganhos

proporcionais, integrais e derivativos que busquem performance e robustez adequadas e os mostra na tela, para que o usuário se preocupe apenas em atingir os padrões de desempenho que ele desejar.

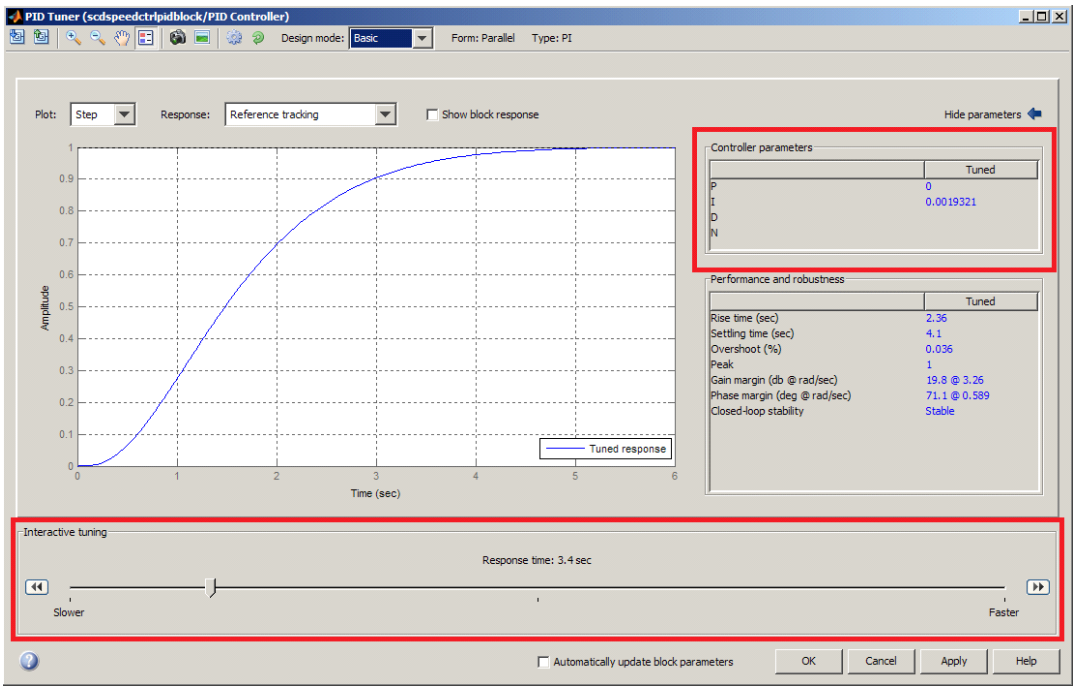


Figura 9 - Ferramenta de auto-ajuste do Simulink®

Para a simulação das malhas de controle, pode-se, inicialmente, partir do modelo construído até o momento no *Simulink*®. Esse será o primeiro modelo de motor explorado. Após a implementação da eletrônica de potência e do circuito de chaveamento, é possível descrever o modelo de forma mais completa, inserindo, no mesmo, elementos correspondentes de eletrônica de potência.

A Figura 10 demonstra uma simples realimentação feita no modelo. É importante notar que nessa realimentação da corrente já está inserido um bloco PI. Esse bloco possui uma ferramenta de auto-ajuste que será aplicada na obtenção dos ganhos para a malha de controle. Outra função desse bloco é a saturação. Esse detalhe será explorado mais adiante no texto, porém é basicamente a possibilidade de limitar a saída da ação de controle para condizer com as limitações reais do projeto.

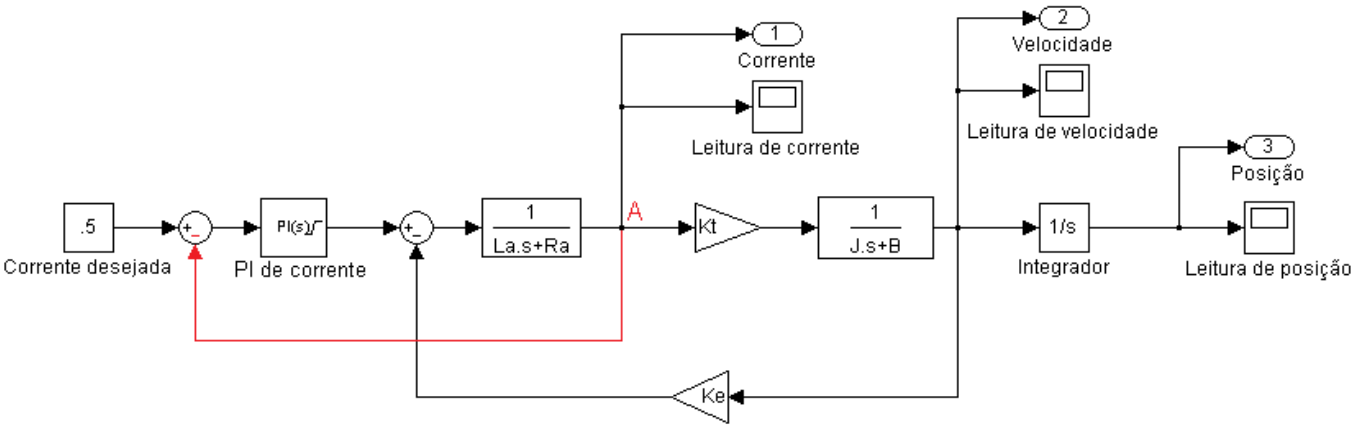


Figura 10- Realimentação de corrente

A realimentação da corrente é necessária para que não exista nenhum pico de corrente indesejado. Isso significa uma maior proteção tanto para o motor como para a eletrônica envolvida no acionamento. No caso do motor, altas correntes podem danificar seu enrolamento, podendo levar até à sua completa inutilização. No caso da eletrônica, existem duas razões para o controle da corrente. A primeira está relacionada à sensibilidade natural dos componentes, que podem vir a se desgastar mais rapidamente. A segunda razão está ligada ao sensor *Hall* de corrente. Como pode ocorrer a saturação desse sensor quando seus limites especificados são ultrapassados, erros de medição e, consequentemente, perdas na ação de controle podem vir a ocorrer. Desse modo, com a corrente controlada, pretende-se evitar tais problemas. Além da função de proteção dos componentes eletrônicos, ainda existe o problema das baterias. Dependendo do tipo de bateria e de sua carga nominal, períodos de sobrecorrente tendem a causar um desgaste maior que o esperado e, portanto, limites devem ser estabelecidos com o propósito de poupar as baterias.

Após ser realizada a realimentação da corrente, para o próximo passo, é necessário realizar o ajuste dessa malha. Esse processo pode ser feito, como já descrito, utilizando-se a ferramenta de auto-ajuste do *Simulink*®. Para isso, é preciso selecionar bloco de controle PID, exibindo as suas propriedades, como mostrado na Figura 11. A seguir, deve-se selecionar, como tipo de controle, o Proporcional-Integral (PI). Selecionando agora, a aba *"PID Advanced"* é possível saturar a saída do bloco PID (Figura 12). Apesar de existir essa possibilidade, no modelo será inserido um saturador

externo, com o intuito de facilitar a visualização. Em seguida, na aba inicial do bloco, devem-se aplicar as mudanças realizadas e selecionar a opção "*Tune...*". Uma nova janela, semelhante àquela mostrada na Figura 9, deverá aparecer, porém com a resposta da corrente do motor CC. Após escolher um valor para ganho que contenha *overshoot*, ou sobressinal, nulo, mas juntamente com um tempo de resposta satisfatório, em torno de 1 segundo, o auto-ajuste pode ser fechado, aplicando os valores encontrados no bloco.

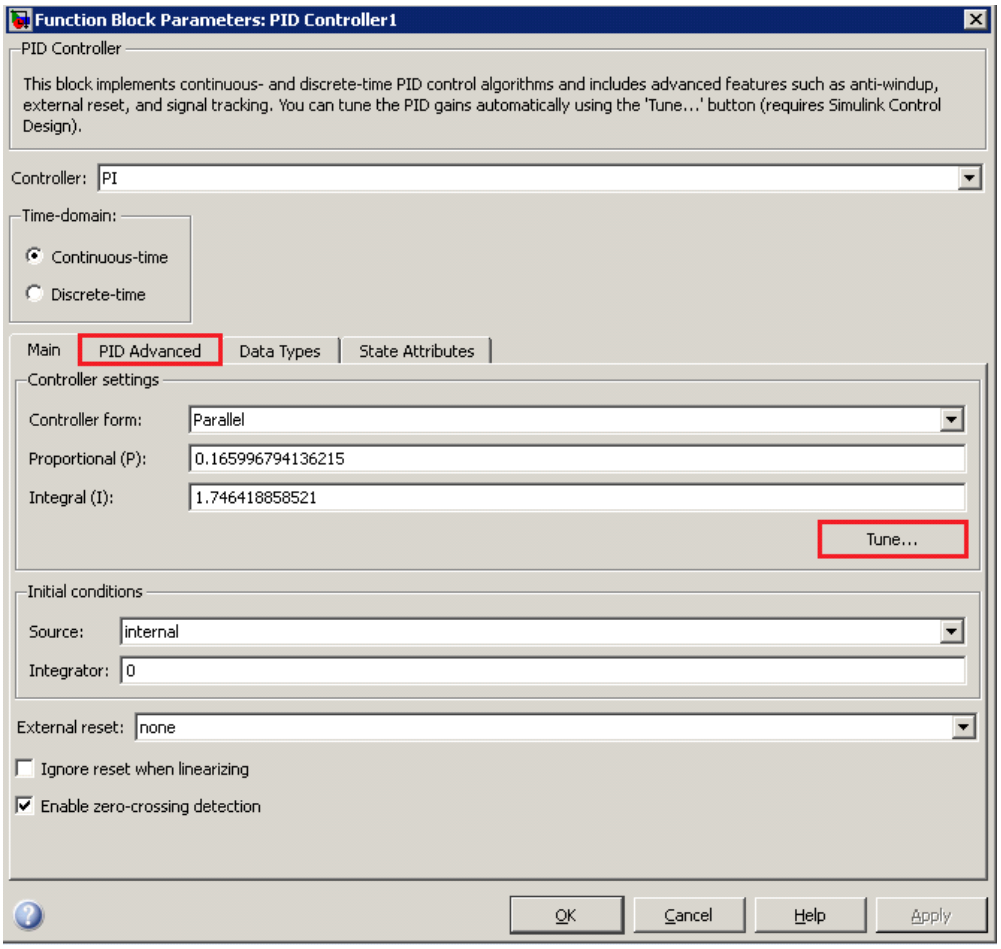


Figura 11 - Tela inicial do bloco PID

O próximo passo deverá ser a realimentação da velocidade (Figura 13). Para esse caso, deve-se acrescentar novamente o bloco PID, do mesmo modo que foi feito com a corrente. A realimentação deve vir do local da planta correspondente ao sinal de velocidade e ser feita com relação à velocidade de referência. Novamente, deve-se usar a função de ajuste automático do PID para a escolha de ganhos que também evitem o sobressinal, enquanto a velocidade possua um tempo de resposta um pouco mais lento

que a corrente. Assim, evita-se que o sistema se comporte lentamente, enquanto as malhas mais exteriores agem mais expressivamente que as malhas mais internas.

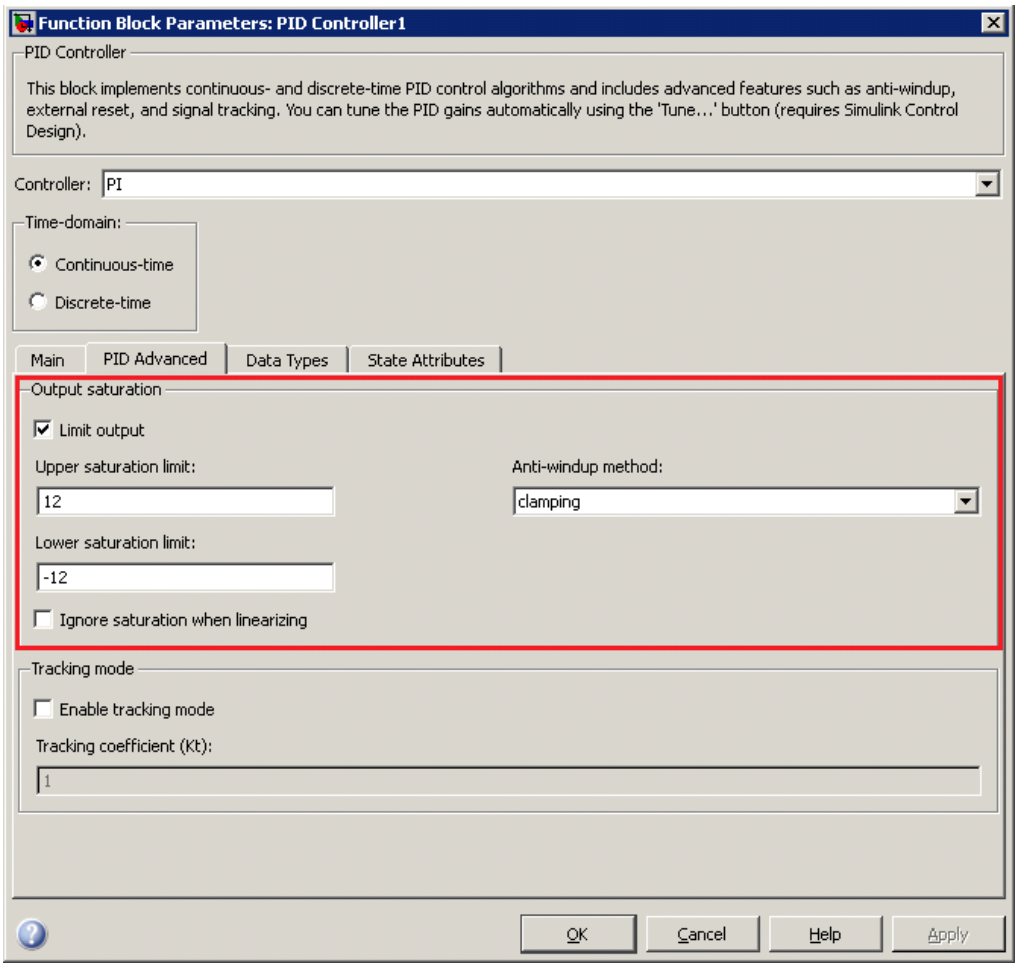


Figura 12 - Saturação da saída do bloco PID

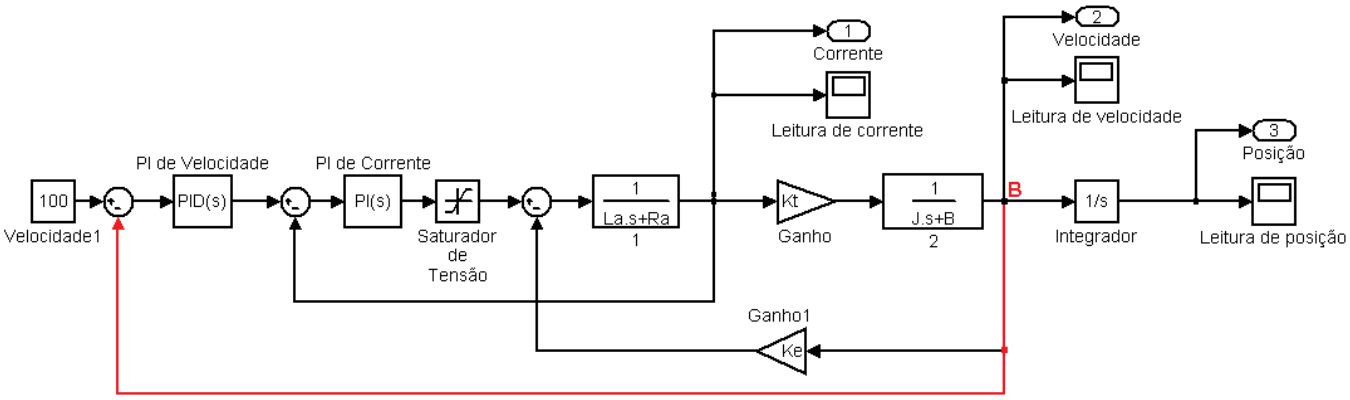


Figura 13 - Realimentação de velocidade

Finalmente, para concluir o objetivo, é necessário realimentar a posição do motor. Para isso, liga-se o bloco integrador à velocidade do motor, tendo a posição como resultado. Esse sinal resultante deve realimentar o sistema modelado do motor, novamente passando pelo bloco somador e pelo bloco PID, cuja saída será a referência de velocidade, completando a malha. O modelo completo do controle do motor está evidenciado na Figura 14.

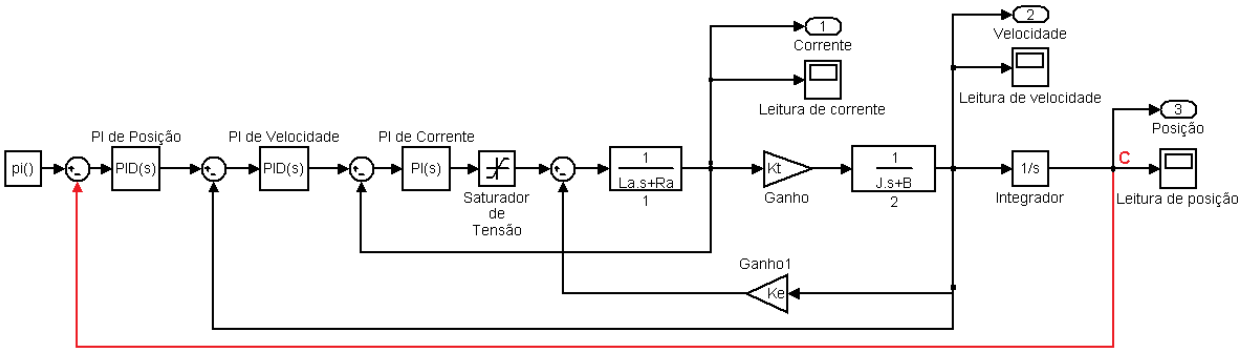


Figura 14 - Realimentação de posição

2.5 Acionamento do motor

No capítulo anterior, foi discorrido sobre a modelagem do motor para o seu controle. Neste tópico, o foco do trabalho estará na elaboração do circuito de acionamento do motor, que deverá obedecer alguns requisitos descritos no decorrer do texto.

O primeiro requisito é com relação com a topologia de acionamento, que deverá ser capaz de acionar o motor tanto no sentido horário, quanto no sentido anti-horário. Esse tipo de operação requer que ora o primeiro terminal do motor esteja ligado à fonte e o segundo ao terra, ora o primeiro esteja ligado ao terra e o segundo à fonte.

O segundo requisito diz respeito à potência que deverá ser entregue ao motor. De acordo com esse critério, o acionamento deve ser não apenas ser robusto o suficiente para suprir, no mínimo, uma tensão de 12V com corrente contínua de 4A, mas também deve ser capaz de controlar a potência entregue ao motor, de acordo com a saída requisitada pelo controle.

O terceiro e último requisito está relacionado com a proteção do circuito. Isso significa que devem existir dispositivos que garantam o desarmamento do acionamento no caso de algum problema na saída que não possa ser detectado pelo controle digital.

Pode-se dizer, portanto, que se trata da implementação de proteção por *hardware*.

2.5.1 Topologia selecionada

Para a seleção da topologia, é preciso considerar os dois primeiros requisitos mencionados anteriormente. A princípio, como se deseja controlar a potência entregue ao motor, é possível verificar que será necessário um conversor CC-CC, mais precisamente, um "*chopper*" (RASHID, 2003).

A operação do motor também influencia na escolha do conversor. Para isso, é possível definir, em um plano "Torque x Velocidade", quatro regiões de operação do motor CC, como mostrado na Figura 15.

No primeiro quadrante, a máquina opera com velocidade positiva e com torque positivo, sendo que ocorre aceleração. No quarto quadrante, a máquina opera com torque positivo, porém com velocidade negativa, o que representa uma frenagem. No quadrante três, ocorre aceleração no sentido inverso do primeiro quadrante, ou seja, a máquina acelera no sentido de velocidade negativa. Finalmente, o segundo quadrante simboliza a máquina operando com velocidade positiva, mas com torque contrário, caracterizando uma desaceleração (PATANÉ, 2008).

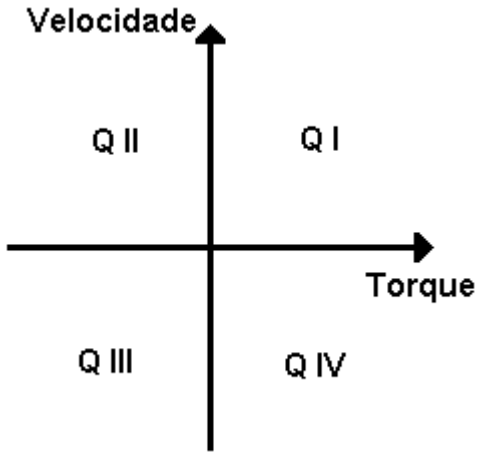


Figura 15 - Quadrantes de operação do motor CC

Além da subdivisão em quadrantes, Rashid também classifica os conversores por classe de operação, sendo de A até E.

Na classe A, a corrente flui para a carga, de modo que tanto a tensão, quanto a corrente são positivas. Essa classe opera no primeiro quadrante.

A classe B considera a corrente como indo para fora da carga, mesmo que haja uma tensão positiva no segundo quadrante. Para uma melhor compreensão, pode-se tomar como referência a Figura 16. Quando a chave CH_1 se fecha, o motor E fornece uma corrente no sentido contrário à i_L, de modo que tem-se, portanto, V_L positiva, com corrente negativa ou saindo do motor.

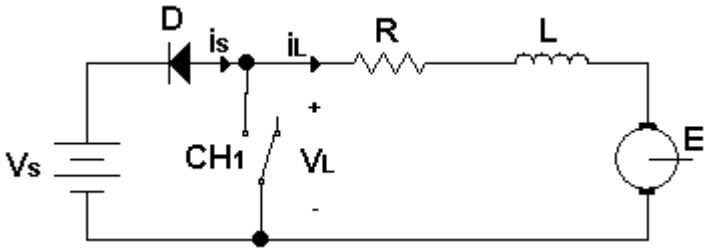


Figura 16 - Chopper classe B

A classe C considera que, na carga, a corrente pode ser tanto positiva quanto negativa dependendo da situação das chaves, ilustradas na Figura 17. Caso a chave CH_1 esteja fechada, a corrente flui da fonte para a carga com a mesma polaridade que V_L. Porém, se a chave CH_1 estiver aberta e CH_2 fechada, a corrente tende a fluir do motor para RL, com polaridade contrária à V_L. Nesse caso, pode-se verificar a condição de operação assim como a da classe B. Esse tipo de operação une tanto a classe A como a classe B.

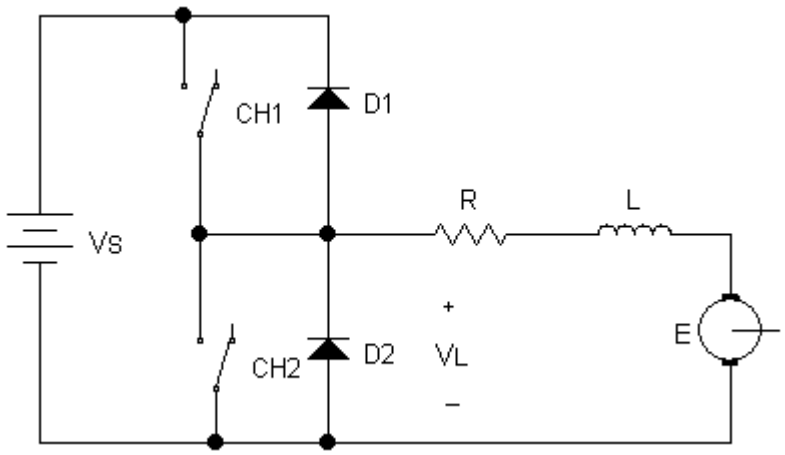


Figura 17 - Chopper operando no primeiro e segundo quadrantes

A seguir, tem-se a classe D, onde a corrente na carga é sempre positiva, sendo que o que varia é o ponto de aplicação da tensão. Isso pode ser verificado analisando o circuito mostrado na Figura 18. Nesse caso, se as chaves CH_1 e CH_2 estiverem fechadas, a corrente flui da fonte para a carga, passando pelas chaves, polarizando o indutor, assim como ilustrado na figura. Com relação à abertura das chaves, a corrente i_L continuará

positiva, um caminho será formado através dos diodos e V_L será invertida.

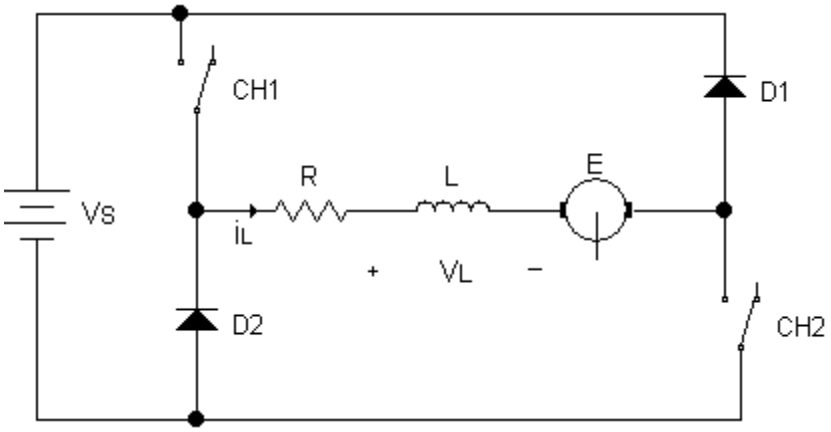


Figura 18 - Chopper operando no primeiro e quarto quadrantes

Por fim, com o *chopper* de classe E, a corrente da carga é tanto positiva como negativa. Semelhantemente, a tensão da carga também pode ser ou positiva ou negativa. Por esse motivo, essa classe de *chopper* também é conhecida como a de quatro quadrantes, que se dá por meio da combinação de dois *choppers* classe C. Uma informação interessante é que, dependendo do modo de operação, este inversor pode ser considerado um monofásico em ponte, dependendo do modo de sua operação. O circuito que demonstra o *chopper* de classe E está mostrado na Figura 19.

Uma vez que atende a todos os requisitos para o controle do motor, a classe E será, portanto, a topologia escolhida para a aplicação de posicionamento.

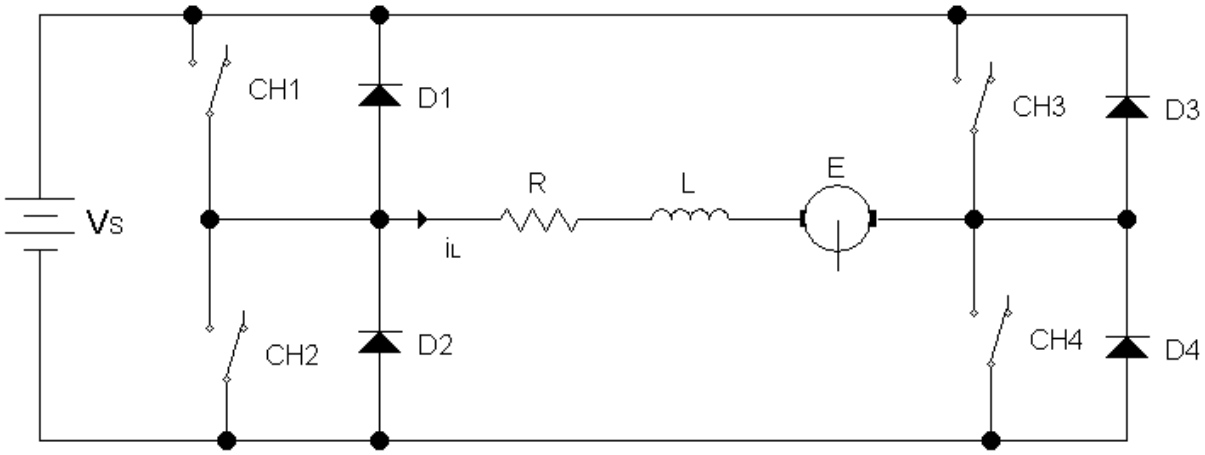


Figura 19 - Chopper de quatro quadrantes

2.5.2 Simulação com o circuito de acionamento

Dada a escolha da topologia de acionamento, é possível agora refinar o modelo

das simulações. Esse é um importante passo para tornar a simulação ainda mais próxima do modelo real de controle do motor. Para isso, foi novamente utilizado o *software Matlab*® com a biblioteca *SimPowerSystems* do *Simulink®,* que contém modelos de motor CC e de pontes de MOSFET's para a simulação do controle.

Partindo do modelo de motor levantado anteriormente, é necessário parametrizar a máquina, disponível na biblioteca do *Simulink*®, de acordo com a Figura 20. Para isso, deve-se inserir a máquina no plano de trabalho, abrir suas opções e selecionar a aba "*Parameters*". No primeiro campo, devem-se colocar os valores de resistência e indutância da armadura; no segundo campo, insere-se a constante de torque "Kt"; no terceiro campo, a constante de atrito viscoso; finalmente, no quarto campo, é o torque necessário para vencer a inércia.

É possível adquirir quatro sinais na saída do modelo do motor CC porém, a corrente de campo torna-se irrelevante, já que o motor a ser simulado é de imã permanente. Para isso, um demultiplexador deve ser colocado na saída do motor com a seguinte ordem de variáveis de saída:

1. Velocidade (wm) em rad/s;
2. Corrente de armadura (ia) em ampères;
3. Corrente de campo (if) em ampères (somente em motores com enrolamento de campo);
4. Torque elétrico em N.m.

A construção do modelo deve prosseguir colocando-se o bloco "*Universal Bridge"*, que é a ponte de MOSFET's, cujos parâmetros são mostrados na Figura 21. Nesse caso, apenas o número de braços deve ser modificado, com o propósito de representar o acionamento proposto.

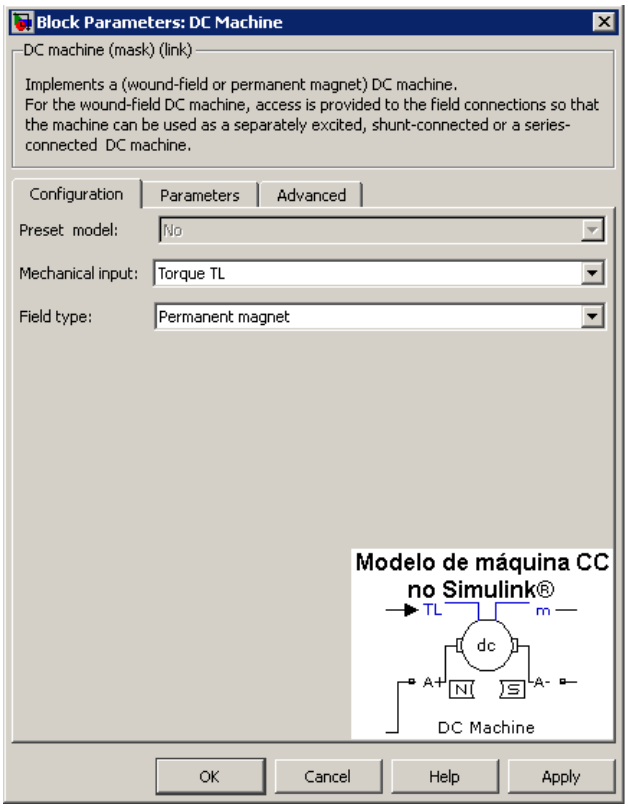


Figura 20 - Parametrização do motor CC para simulação no Matlab®

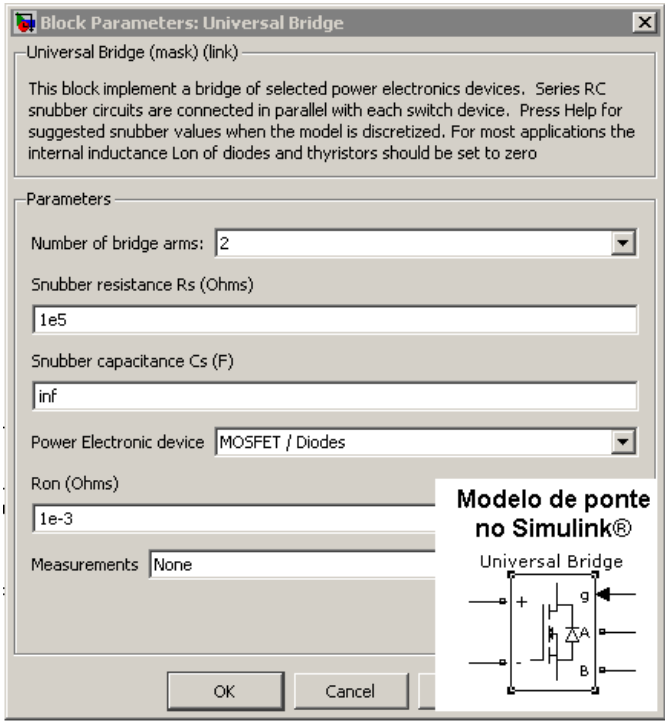


Figura 21 - Parametrização da ponte de MOSFETs

Uma vez que os sistemas básicos já foram inseridos na planta, o próximo passo é a ligação entre os componentes da planta. Para simplificar a visualização do modelo no *Simulink*®, foi criado um subsistema que recebe os sinais de velocidade, de corrente do motor e de referência. Tal subsistema também contém, internamente, blocos para o controle da máquina e, em sua saída, é computado um valor para ser sinalizado ao PWM, que gera os pulsos para a ponte inversora.

A planta completa, incluindo o subsistema, pode ser vista na Figura 22. Diversos sinais estão sendo medidos com a utilização da ferramenta "*Scope*", além de serem exportados para a área de trabalho *Matlab*®, onde podem ser trabalhados e avaliados mais detalhadamente.

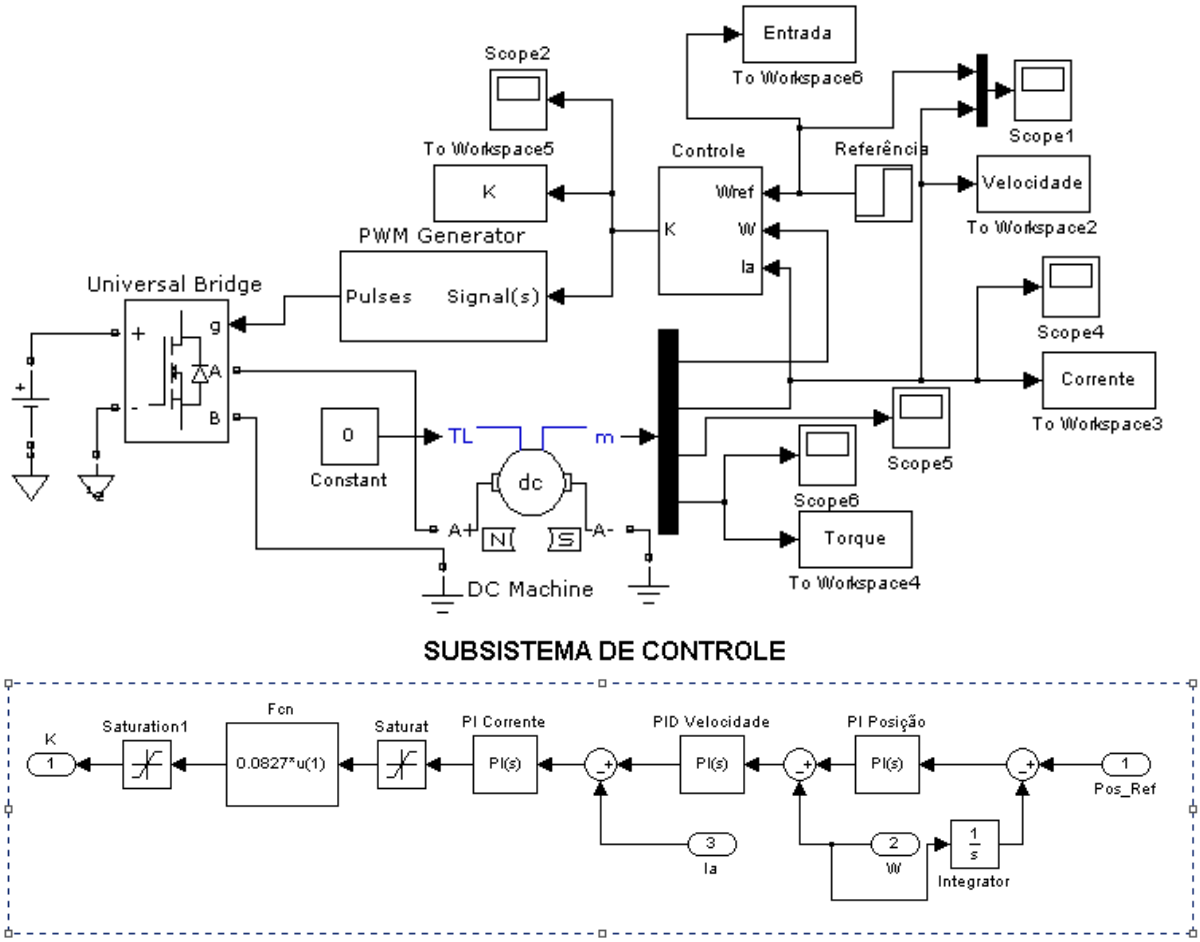


Figura 22 - Simulação completa do sistema de potência e controle

2.5.3 Desenvolvimento da placa de acionamento do motor

O circuito elétrico que será responsável pela alimentação do motor terá, como base, o esquema mostrado, anteriormente, na Figura 19. As chaves são substituídas por

transistores, o motor é a carga e o comando para acionamento deve vir do microprocessador. O esquema dessa topologia, já adaptado à aplicação, segue ilustrado na Figura 23:

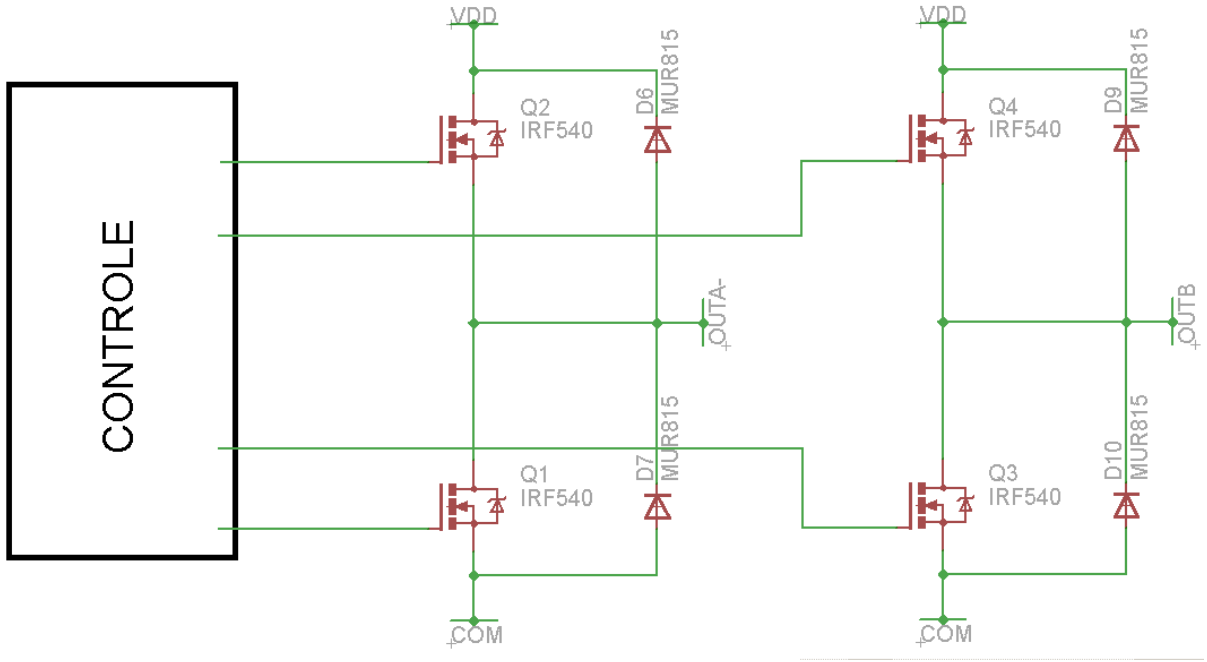


Figura 23 - Esquema de chopper de quatro quadrantes utilizando transistores

A habilitação dos MOSFET's deve vir do microprocessador e é preciso ter cuidado com a isolação da tensão entre o controle e o acionamento. Para isso, é necessária a utilização de opto-acopladores. Esses dispositivos têm como função a isolação dos sinais de entradas e saídas e devem ser utilizados para adequar diferentes tensões usadas em circuitos. Um exemplo é o do circuito de acionamento que trabalha na faixa de tensão de 0 a 12 volts, enquanto o circuito de controle trabalha na faixa de 0 a 5 volts.

Para o acionamento dos transistores propriamente ditos, pode ser utilizado um circuito integrado (CI) que já contenha proteções internas e externas por *hardware* que auxiliarão na prevenção de falhas. Um CI que faz o papel de *driver* para acionamento dos *gates* dos transistores é o IR2112. Esse CI é capaz de acionar o transistor e possui uma entrada específica para desligamento das saídas que pode ser utilizada em caso de emergência. Além disso, esse *driver* possui duas entradas (HIN e LIN), que servem como sinal para o acionamento dos *gates* do MOSFET, como mostrado na Figura 24:

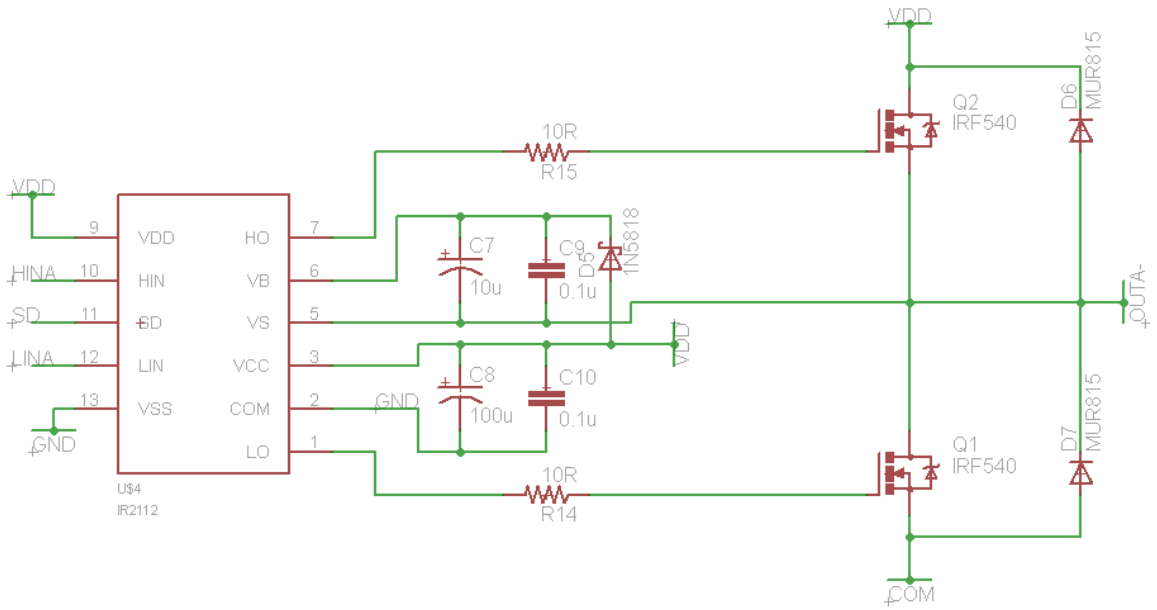


Figura 24 - Esquema de ligação do driver dos MOSFET's

O sinal de *"Shutdown"* (SD) deve ser proveniente de uma lógica que permita o desligamento por *software* e também por *hardware*, como no caso de um evento emergencial, como um curto-circuito, por exemplo. Essa lógica é construída a partir de dois sinais: "SC" e "ENABIN" negado.

A Figura 25 mostra o circuito utilizado para a proteção por *hardware* no caso de sobrecorrente ou curto-circuito dos terminais de saída do inversor, o qual terá como saída o sinal SC. No circuito elétrico utiliza-sa um resistor *shunt* (R_S), ligado entre a saída do motor (COM) e o terra (GND). Quando a tensão sobre o *shunt* ultrapassa 0.8 volts, o que representaria 8 ampères, o transistor T2 entra em condução, bloqueando a condução de T1 e fazendo com que a saída SC apresente a tensão V_{DD}.

O desligamento por *software* complementa a lógica de desligamento por *hardware* no circuito mostrado na Figura 26. Quando o sinal de nível lógico "0" vem do microprocessador, a base do transistor do opto-acoplador é acionada, fazendo com que, no CI 4093N, portas NAND *schimitt-trigger*, a saída seja "1". Caso o sinal de "SC" também seja "0", a saída de desligamento (SD) será desativada e, desse modo, a ponte poderá ser acionada.

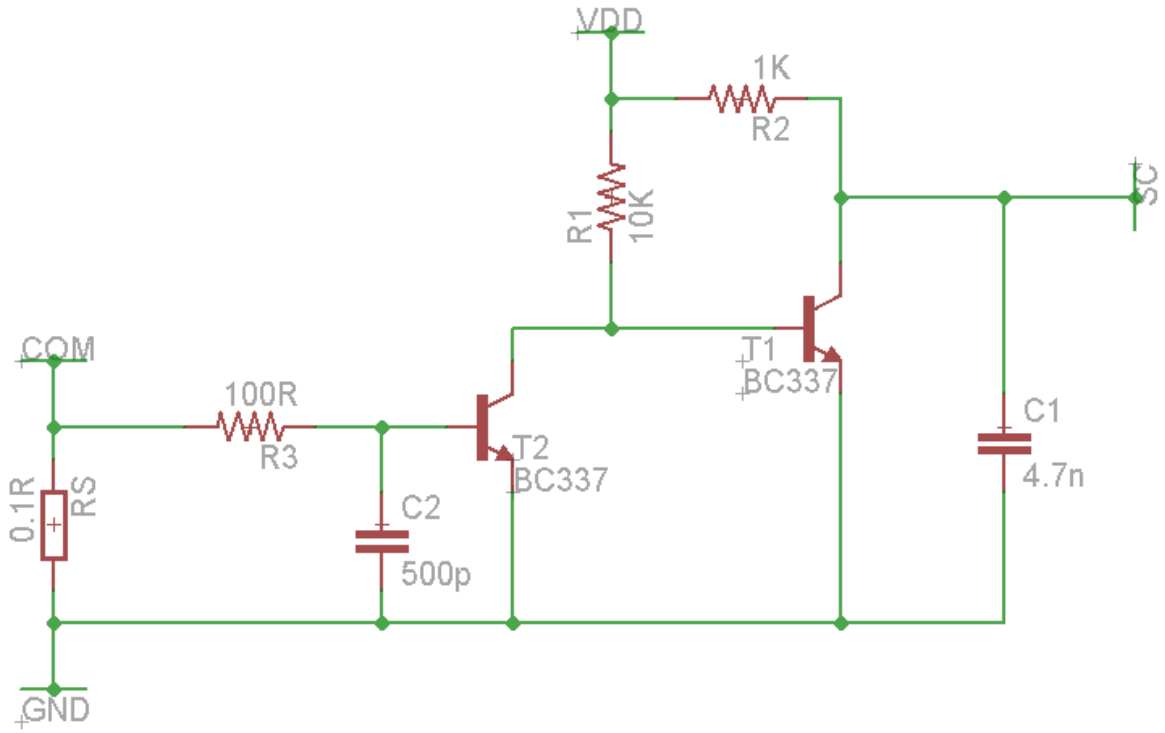


Figura 25 - Esquema de proteção de curto-circuito

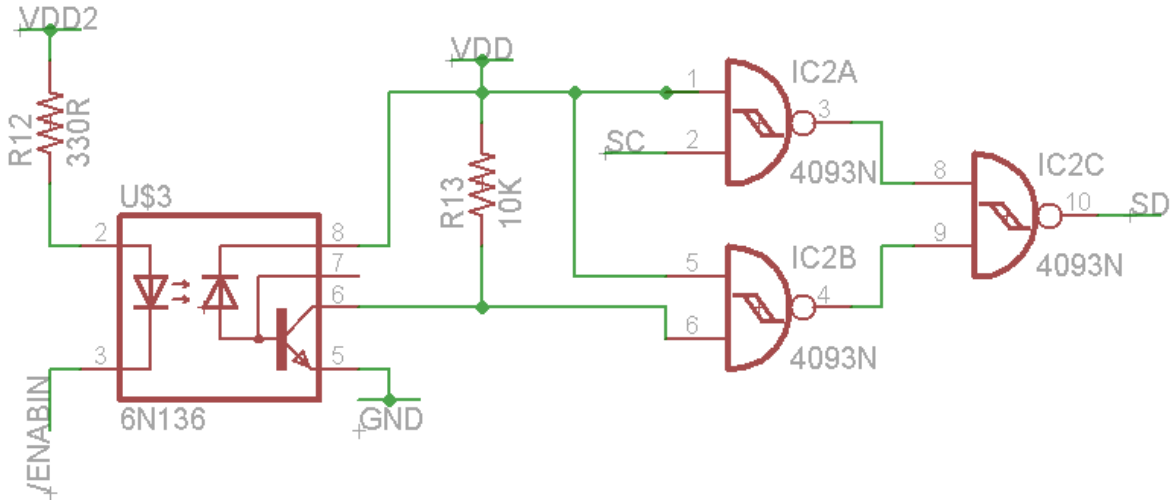


Figura 26 - Esquema da saída do sinal de desligamento da ponte

Ainda tendo como base o circuito de acionamento, mostrado na Figura 24, os próximos sinais necessários para que o motor seja alimentado são “HIN” e “LIN”. Neles, devem chegar os pulsos provenientes do microcontrolador, que controlam a tensão aplicada aos terminais do motor. Para a obtenção dos pulsos de controle, foi montado o esquema mostrado na Figura 27. Novamente foi utilizado um opto-acoplador para isolar o sinal proveniente da placa de controle, do sinal que vai até o acionamento. No caso da entrada ter nível lógico “0”, o sinal “LIN” estará em “1” e “HIN” em 0, e vice-versa.

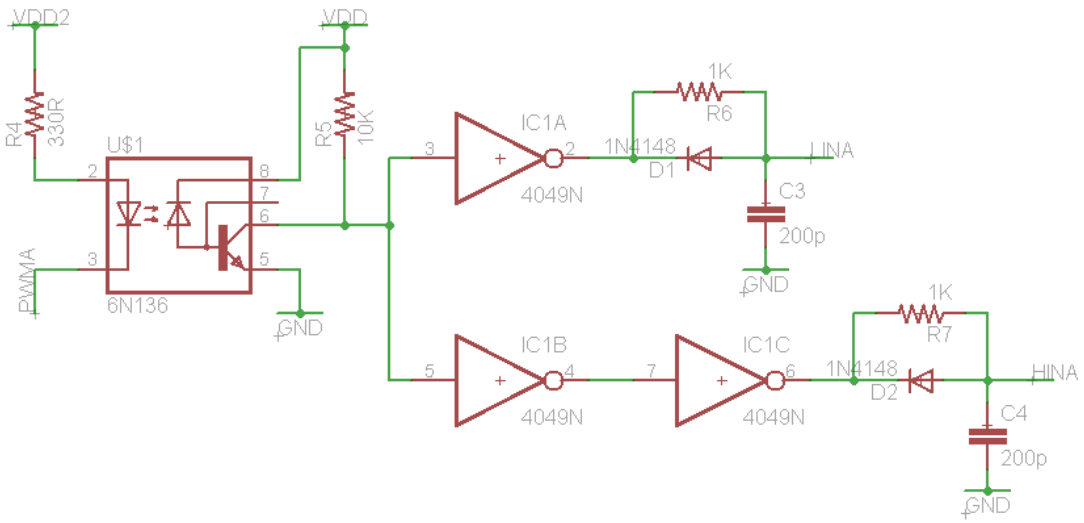


Figura 27 - Adequação do sinal de entrada para controle do acionamento

O último módulo a ser integrado à placa de controle é o sensor de corrente por efeito *Hall* ACS712T. Esse dispositivo tem como propriedade, a mudança na tensão de saída de acordo com a variação do campo magnético que o atravessa. A integração desse sensor na placa de potência permite a leitura de corrente, limitada a 5 ampères, de qualquer dispositivo conectado à saída de potência uma vez que a saída do sensor já é proporcional tanto a correntes positivas como negativas.

2.6 Aplicação do microprocessador ARM Cortex-M3

O último passo do desenvolvimento do controle do motor está na integração de todos os sistemas até aqui descritos. Essa integração é realizada por meio da interface com o ARM Cortex-M3, um microprocessador da família Cortex-M, especialmente desenvolvido para ser aplicado como microcontrolador (YIU, 2009).

A integração do microprocessador ARM e o meio externo é feita utilizando uma placa de desenvolvimento do fabricante *Olimex*, a STM32-H103 (Figura 28). Essa placa já contém diversos circuitos auxiliares que facilitam o desenvolvimento de diversas aplicações, pois já contém circuitos de *reset*, osciladores, pinos para acesso das portas de entrada e saída e também uma porta USB, que além de trocar dados, permite alimentar todo o circuito (OLIMEX, 2008).

O núcleo da placa de desenvolvimento da *Olimex* é um ARM Cortex-M3 da fabricante *STMicroelectronics*, o STM32F103RBT6. A Tabela 2 mostra algumas das especificações desse microprocessador:

Tabela 2 - Especificações do STM32F103RBT6

Clock da CPU	Até 72MHz
Flash	128 kB
RAM	20 kB
DMA	7 canais
Real Time Clock	1
Watchdog Timer	1
Timers	4
SPI	2
I2C	2
USART	3
USB	1
CAN	1
GPIO	51
ADC	2

Além dessas características, deve-se notar o baixo consumo de energia da placa toda, que, com a utilização de todos os periféricos e em velocidade máxima, fica em torno de 40mA e a possibilidade de operação com uma tensão uma tensão aplicada que varie de 2V no mínimo, até 3.6V, no máximo. Outro fator a ser considerado é a possibilidade do processador de entrar em modos de espera, reduzindo ainda mais o consumo do circuito (OLIMEX, 2008).

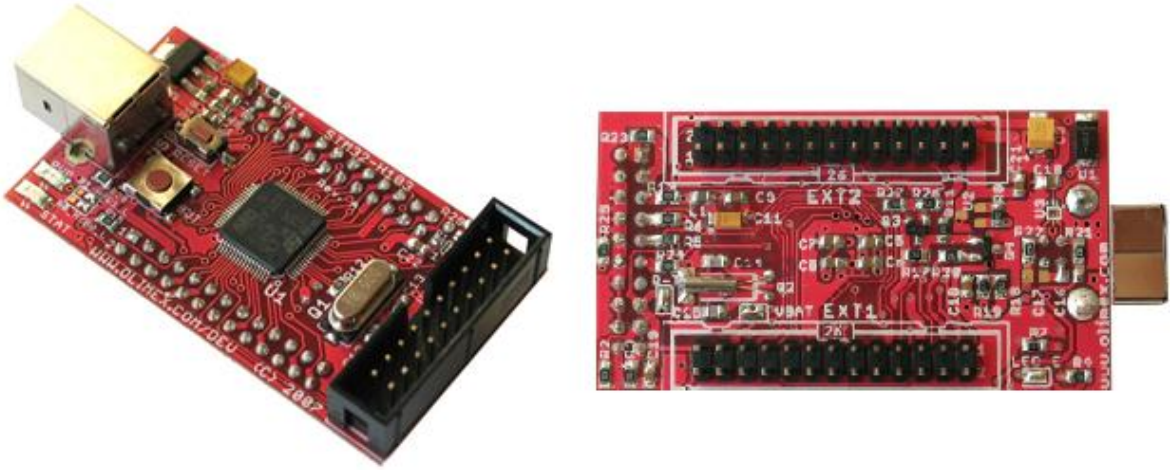


Figura 28 - Placa de desenvolvimento da *Olimex*

Um dos problemas que devem ser contornados é o da tensão de entrada e saída dos periféricos da placa de desenvolvimento STM32-H103. O microprocessador ARM está limitado a um máximo de tensão de 3.6V em suas portas, mas os periféricos se comunicam com níveis de tensão maiores. Desse modo, é necessário o desenvolvimento de uma placa de interface que adeque os sinais de entradas e saída evitando a queima do microprocessador.

2.6.1 Desenvolvimento da placa de interface

Primeiramente é preciso realizar um estudo para o mapeamento das portas disponíveis na placa STM32-H103 e posterior seleção dos pinos que serão utilizados. Para isso, deve-se utilizar o manual do usuário (OLIMEX, 2008) que contém uma figura que mostra alguns periféricos mapeados em cada pino do circuito de desenvolvimento. Para o caso de aplicação de controle de motores, foram escolhidos sinais conforme mostrados na Tabela 3 e a Figura 29 mostra a pinagem utilizada para esses sinais.

Tabela 3 - Sinais para interface

Conversor A/D	5
Entradas Digitais	6
Saídas Digitais	6
Conversor D/A	1

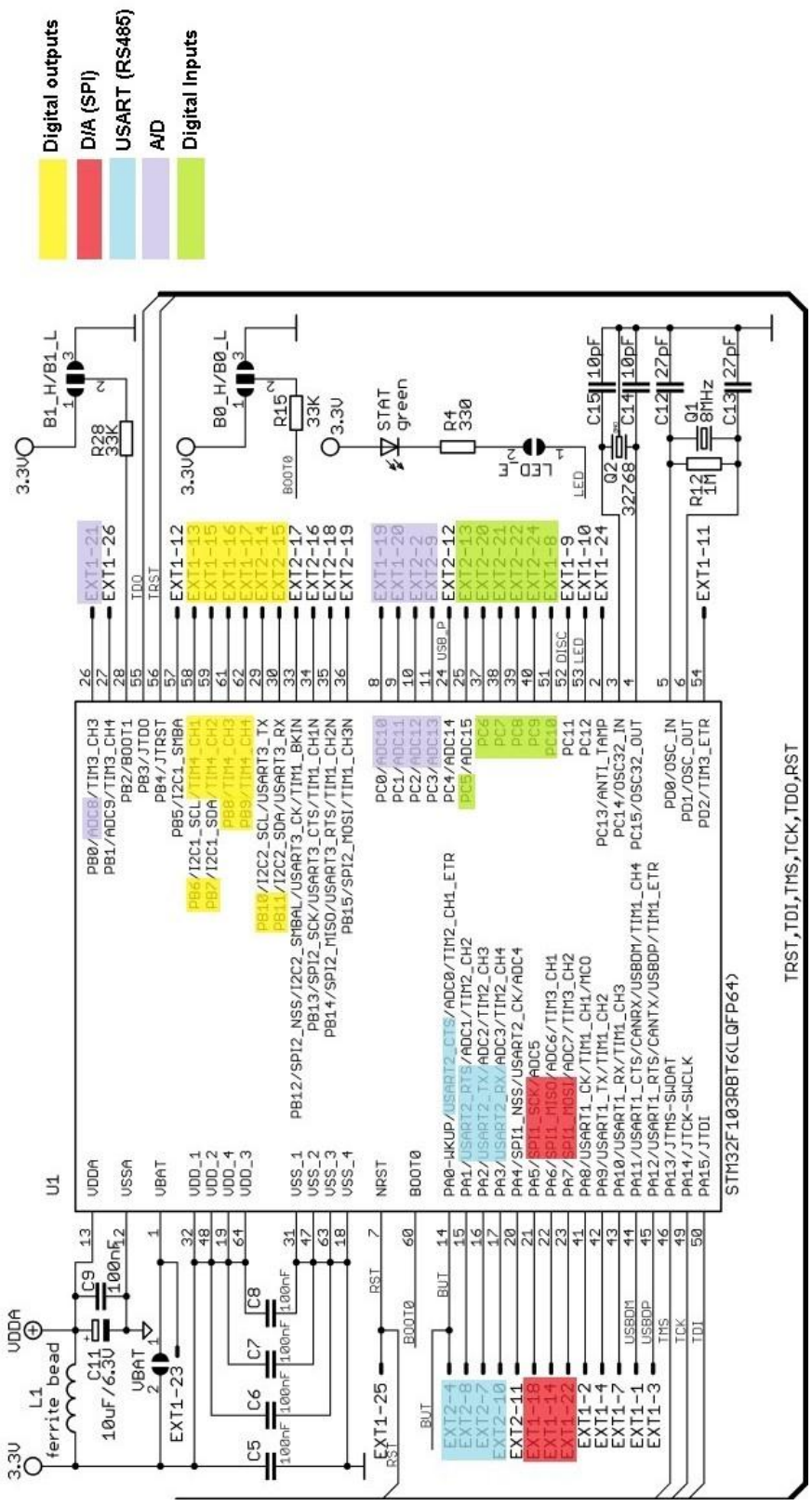


Figura 29 - Diagrama de utilização do ARM

Os sinais previstos têm como finalidade a aplicação da placa de interface para objetivos além do controle do motor. Isso permite que o controlador possa ser inserido em qualquer aplicação, desde que sejam observados os limites de tensão pré-estabelecidos, e que já esteja preparado para, se necessário, realizar o controle de mais de uma atividade ao mesmo tempo. No presente trabalho, a placa de interface está totalmente voltada à aplicação de acionamento do motor CC.

Para a utilização dos sinais digitais de entradas e saídas do microprocessador, foram utilizados circuitos integrados capazes de adaptar os níveis lógicos de operação, de 0 a 3.3V para 0 a 5V, que foi o padrão escolhido da interface. Para as entradas digitais, foi utilizado o CD4050BC, e, para as saídas digitais, foi utilizado o CI 74HCT541, que também atua como *buffer* de saída, permitindo a drenagem de maior corrente e, consequentemente, o acionamento de mais dispositivos. O circuito desenvolvido para a aplicação dessas duas adequações encontra-se na Figura 30.

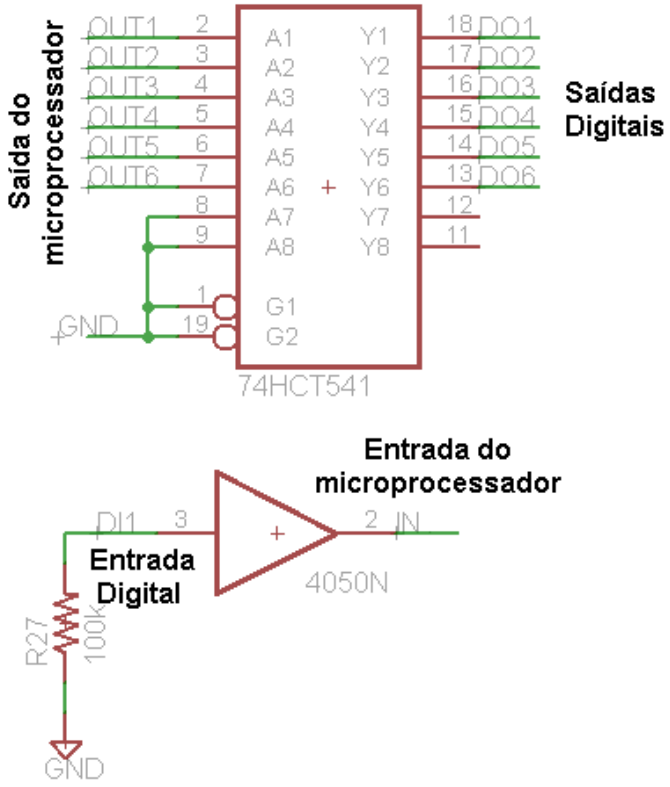


Figura 30 - Circuitos para adequação de sinais digitais

O circuito para o conversor A/D também deve ser construído de modo a aceitar tensões de entrada que variem de 0 a 5V, no modo diferencial. Para isso, foram utilizados amplificadores operacionais com ganho de 0,5, com o propósito de facilitar os cálculos a serem realizados pelo microprocessador, uma vez que o seu valor convertido é sempre metade do valor medido.

Um detalhe importante a respeito do circuito de adequação dos sinais do conversor A/D é o fato de que não está disponível uma alimentação simétrica (-12V - +12V). Por esse motivo, o amplificador operacional escolhido para a aplicação foi o LMC6062, devido à sua característica *rail-to-rail* - nome dado quando o amplificador operacional consegue excursionar a sua saída entre níveis próximos aos da sua alimentação (BRAGA, 2009). O circuito de entrada do conversor A/D do microprocessador está mostrado na Figura 31:

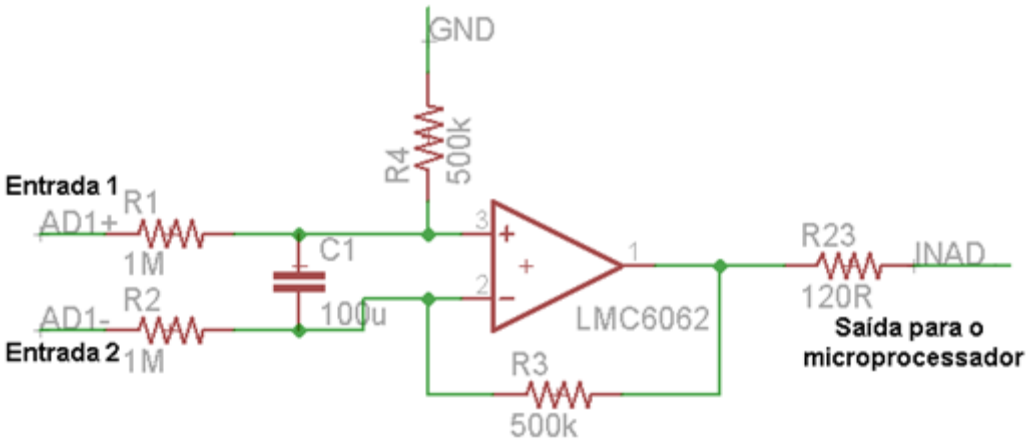


Figura 31 - Circuito de entrada de sinais analógicos

Para a utilização do conversor Digital-Analógico do microprocessador, foi utilizado um dispositivo compatível com o protocolo SPI (*Serial Pepipheral Interface*). O dispositivo em questão é o MCP4921. Esse circuito integrado traduz o valor digital, que é enviado através de uma comunicação serial, para uma saída de tensão proporcional. A desvantagem da utilização dessa interface na placa STM32-H103 se dá pelo fato de que o barramento responsável pela comunicação SPI também é compartilhado pelo barramento do depurador. Portanto, caso seja usada a interface SPI, não é possível realizar a depuração em tempo real. As conexões do conversor A/D estão mostradas na Figura 32.

Para a comunicação do ARM com outros dispositivos, está prevista a implementação de uma interface para rede serial utilizando o padrão RS485. O circuito integrado utilizado para a troca de sinais é o MAX 485 e, juntamente com o esquema, foi adicionado um *jumper* com o intuito de sinalizar o fechamento da rede, como pode ser visto na Figura 33:

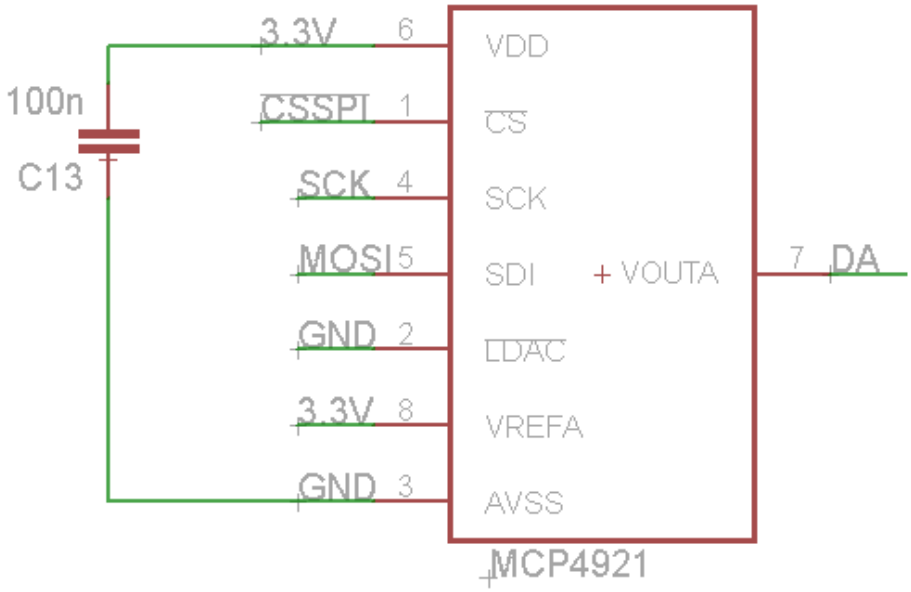


Figura 32 - Ligação do circuito do conversor D/A

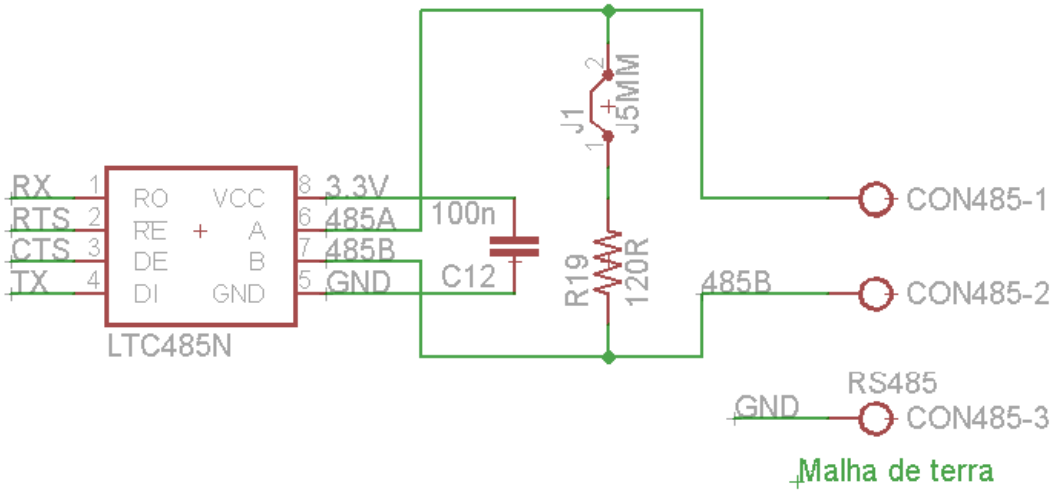


Figura 33 - Interface do microprocessador por RS485

A alimentação do circuito de interface é proveniente de uma fonte de 12V ou de

baterias, sendo que a regulação da tensão para os circuitos internos é feita utilizando-se o CI 7805 para a regulação em 5V e do LM317 para a saída em 3.3V. O esquema de alimentação da placa de interface está mostrado na Figura 34.

Finalmente, para terminar o esquema da placa de interface, é necessário acrescentar as conexões de entradas e saídas no ARM e para o meio externo. Portanto, foram colocados quatro conectores sendo que dois são para o encaixe dos pinos da placa de controle (STM32-H103) e os outros dois para as entradas e saídas. As ligações podem ser vistas na Figura 35.

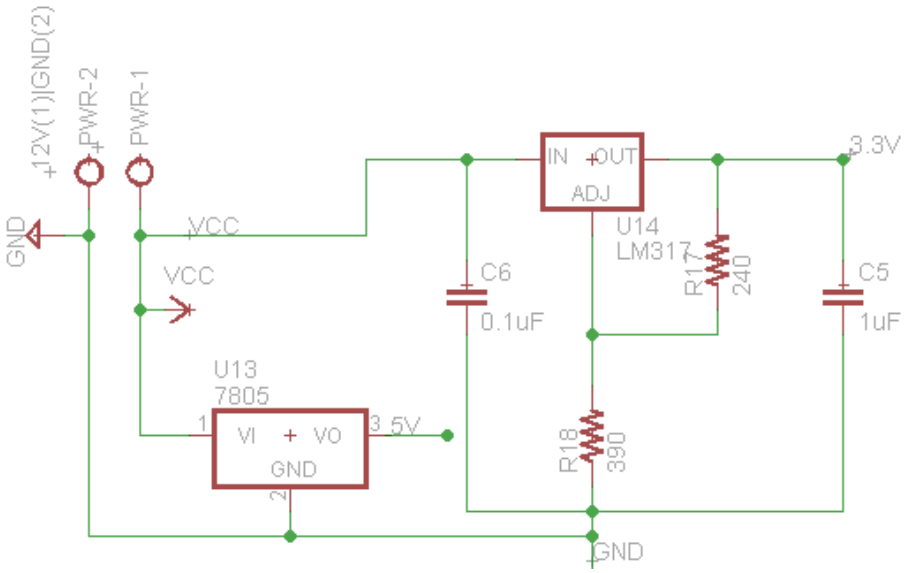


Figura 34 - Ligação para os circuitos de alimentação

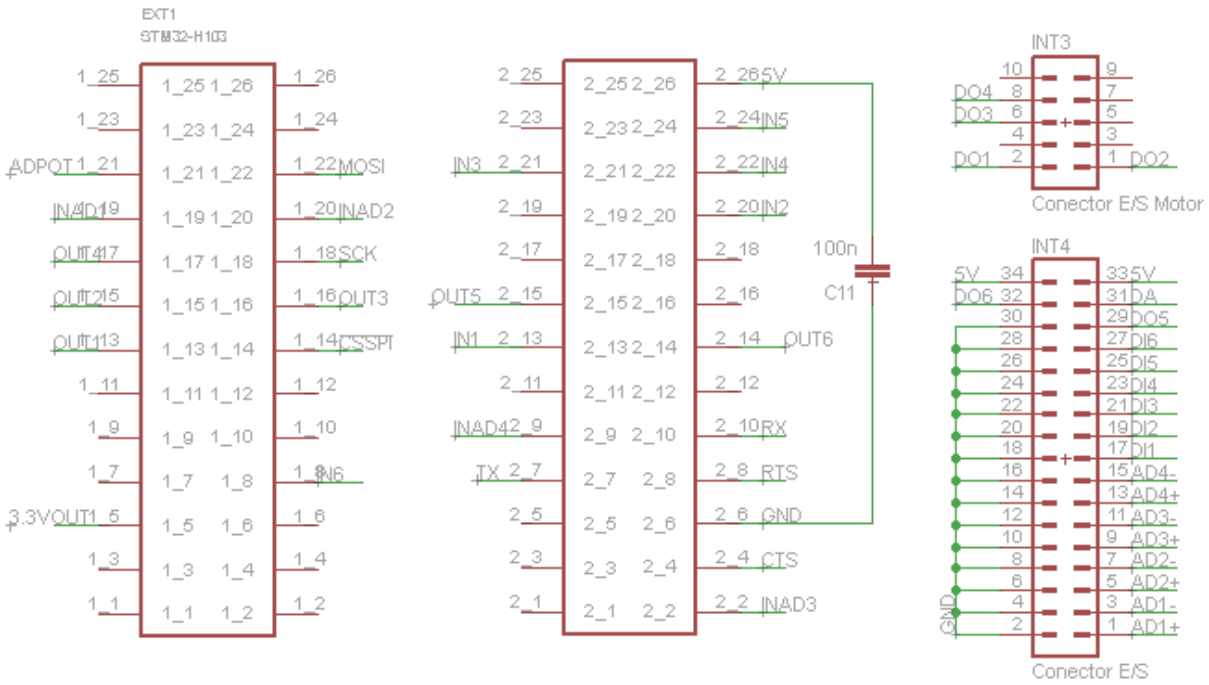


Figura 35 - Diagrama de entradas e saídas

2.6.2 Preparação para a programação do ARM Cortex-M3

Uma das razões para escolha do ARM Cortex-M3 deve-se à sua vasta gama de recursos para programação que variam desde linguagem de baixo nível, como a programação em linguagem *Assembly*, até linguagens de alto nível, através de compiladores, dispondo de variados recursos de depuração e programação por blocos (YIU, 2009).

Para o presente trabalho, optou-se pela programação utilizando ferramentas livres, que são distribuídas gratuitamente na internet. Para a compilação do código foi utilizado o *software* "*arm-elf-gcc*" versão 4.5.2. Para a depuração, foi utilizado o "*arm-elf-gdb*" versão 6.6. A interface gráfica escolhida para a programação e depuração foi o *software* "*Emacs*" versão 23.1.1. Já a comunicação com a placa é feita através do protocolo JTAG, pela porta USB do computador, e para a troca dos dados, utiliza-se o *software* "*OpenOCD*" versão 0.6.6. O sistema operacional utilizado como base para todos os programas citados é o *Linux*, com a distribuição *Ubuntu* na sua versão 10.10 e versão de *kernel* 2.6.35-28-generic.

Inicialmente, é necessário compilar cada um dos *softwares* de acordo com as configurações de *hardware* que serão utilizadas. Para isso, foi utilizado um *script* que já contém todos os parâmetros necessários para o funcionamento e integração dos

softwares. O *script* para a compilação das ferramentas está disponível para *download* por meio de um *link* (HERRMANN, 2009, último acesso em 03/11/2011).

Após a instalação dos programas, ainda é necessário gerar o arquivo de configuração do *OpenOCD*. Esse arquivo contém informações para que o *software* comunique-se com o microprocessador ARM utilizando a interface com padrão JTAG (*IEEE Std 1149.1-1990)*. A configuração foi gerada com base no manual de operação do *OpenOCD* (BROWNELL, DUANE, *et al.*, 2011).

Além das ferramentas já citadas, é preciso escolher uma biblioteca que contenha as definições necessárias para compilar o código desenvolvido. Há diversas opções de bibliotecas para o ARM Cortex-M3 e algumas delas são gratuitas e abertas para desenvolvimento. A opção utilizada é a biblioteca "*Standard Peripheral Library*" desenvolvida pela *STMicroelectronics* especialmente para aplicações que utilizam *CI's* STM32F10x. Existem diversos recursos que auxiliam o desenvolvimento, além de exemplos de aplicações, que podem ser vistos e baixados na página da *web* da empresa STMicroelectronics (STMICROELECTRONICS, 2011).

2.6.3 Desenvolvimento do *software* de controle

O desenvolvimento da rotina de controle no ARM Cortex-M3 foi realizado seguindo o fluxograma mostrado na Figura 36. O microprocessador inicia, após o *reset,* em uma rotina de organização da memória para os dados que serão inseridos na RAM. Em seguida, são feitas as inicializações das variáveis que serão usadas no programa principal e executadas rotinas que inicializam os periféricos do microprocessador, preparando suas portas como entradas ou saídas (YIU, 2009).

Ao final da inicialização de todas as entradas e saídas, começa-se uma rotina de calibração interna dos conversores A/D, uma vez que não há um pino de referência exclusivo para o conversor analógico-digital. Terminada essa rotina, segue-se a calibração do ponto zero de corrente e velocidade. Essa rotina é necessária, pois os sensores são alimentados com fontes que apresentam uma variação natural de tensão e, como dependem de um valor referência maior do que zero, o ponto médio varia. Portanto é necessário encontrar esse ponto a pedido do usuário e também sempre que o módulo de controle inicie sua operação.

Após a calibração do zero dos conversores de corrente e velocidade, libera-se o microprocessador para iniciar o controle do motor. Nesse ponto, a rotina principal é responsável apenas por verificar se o usuário requisitou uma nova calibração do zero de

velocidade e corrente. O controle do motor ocorre dentro da rotina de atualização do PWM. O controle realizado dentro do PWM segue os passos detalhados a seguir:

Primeiramente, quando o período do PWM chega ao fim, é executada a interrupção do *Timer* associado a esse sinal. Dessa forma são desativadas todas as outras interrupções, avaliando se o controle por *hardware* está habilitado, na entrada digital 4. Em caso negativo, a rotina chega ao fim e o motor continua desligado.

Caso o controle esteja habilitado, o valor da posição é lido e é calculado o erro com base no *setpoint* de posição. A partir desse cálculo, é aplicado o algoritmo digital do PI, que será descrito adiante. A saída do PI de posição é a entrada do PI de velocidade, onde, novamente, é calculada a ação de controle proporcional ao erro e a saída desse controlador é a entrada do controlador de corrente.

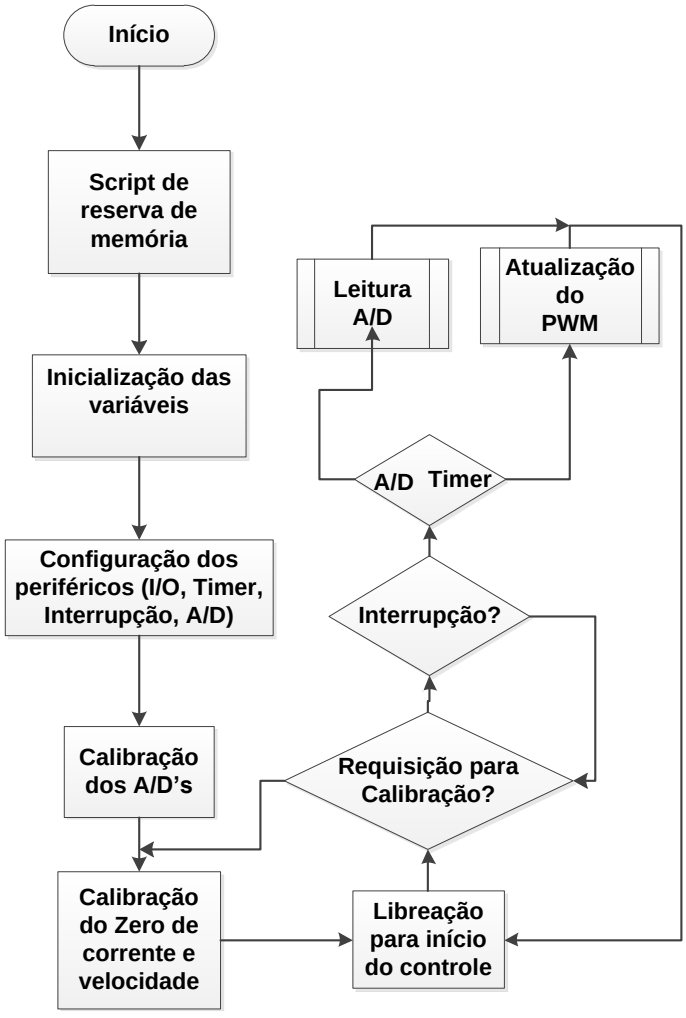


Figura 36 - Fluxograma do *software* de controle

Com todos os cálculos realizados, o último estágio antes da atualização do PWM é a saturação da saída que deverá condizer com a saída real. Para isso, é limitado o valor do PWM de 0 a 100%. Em caso de um valor maior, a saída será saturada em 100%.

Nesse momento, o sinal está pronto para ser atualizado no PWM. O programa então calcula o sentido de rotação do motor. Caso o resultado do PWM seja maior do que 0, um dos lados da ponte é acionado. Caso contrário, deve-se inverter o acionamento da ponte a fim de que o motor inverta também o seu sentido de giro.

Ao final dos cálculos e da determinação do sentido de giro, o valor do PWM é atualizado no registrador referente ao período do pulso, a interrupção de leitura do conversor A/D é ligada novamente e o *bit* referente à interrupção do *Timer* é levado ao nível lógico 0.

É importante observar que a maneira como a biblioteca e o compilador foram preparados, não há suporte para a realização de operações com números reais, somente números inteiros. Porém, existem duas opções caso o trabalhado com valores fracionados for imprescindível. A primeira alternativa é multiplicar os valores trabalhados por múltiplos de 10, realizar os cálculos e dividir o resultado na mesma ordem de grandeza da multiplicação, tendo em mente que serão arredondados os números não inteiros. Outra alternativa é usar a rotação de *bits* à direita. Nesse caso, a cada *bit* rotacionado, o valor é dividido por 2. Ainda assim, tem-se números não inteiros arredondados, todavia, o cálculo será mais rápido se comparado a uma divisão, por se tratar de uma operação com mnemônico único e direto em *Assembly* (YIU, 2009).

2.6.4 Implementação do controle PI digital

Como proposto anteriormente, as malhas de corrente, velocidade e posição devem ser reguladas por controladores PI. Como o controle é feito pelo microprocessador ARM, sua variação é discreta, e não contínua. Desse modo, a abordagem do controle deve passar de contínua para discreta e é necessário fazer a adaptação do controlador da forma contínua para a discreta.

O modelo contínuo do controlador PI pode ser descrito como visto na equação 16 e uma outra maneira de se representar o termo integral, para a adaptação do modelo contínuo para o discreto, está mostrado na equação 17 (THAM, 1998).

$$u(t) = K_C\, e(t) + \frac{K_C}{T_i} \int_0^t e(t) + u_0 \qquad (16)$$

$$\int_0^t e(t)\, dt \approx T_S \sum_{i=0}^{t} e(i) \qquad (17)$$

Onde:

- K_C – Ganho proporcional
- e(t) – Erro calculado da entrada com relação à saída
- T_i – Ganho Integrativo
- T_S – Tempo de discretização

A equação 17 utiliza-se de uma aproximação do comportamento real do sinal por meio de diversas janelas retangulares, discretizando a parte integrativa, assim como é feito quando aplicado o microprocessador. A seguir, fazendo a diferença Δu(t) na equação 16, tem-se como resultado a equação 18 (HAUGEN, 2008):

$$\Delta u(t) = K_C \left\{ [e(t) - e(t-1)] + \frac{\Delta T}{T_S}\, e(t) \right\} \qquad (18)$$

Uma vantagem dessa representação é a supressão da somatória no termo integral, posto que essa somatória pode causar problemas de *overflow* nos registradores. Todavia, agora se torna necessário calcular a saída u(t), como mostrado na equação 19. Isso pode ser feito facilmente armazenando o valor anterior da saída. Portanto, passando esses parâmetros para um código em linguagem C, consegue-se um algoritmo de controle PI digital, necessário ao controle do motor.

$$u(t) = \Delta u(t) + u(t - \Delta t) \qquad (19)$$

3. Resultados

3.1 Parâmetros do motor CC

Inicialmente, foram medidos diversos valores de resistência para diferentes posições do eixo do motor. As medidas foram anotadas, escolhendo-se o menor valor, como proposto no método de determinação dos parâmetros do motor CC para a determinação da indutância do motor. Para a medida da indutância, o eixo do motor foi mantido na posição da medida de menor resistência e, como proposto pelo método, um medidor de indutância foi conectado aos seus terminais. A Tabela 4 exibe os valores de resistência medidos. O valor da indutância da armadura do motor CC medido é 1,983mH.

Tabela 4 - Valores de resistência e indutância de armadura do motor CC

Resistência elétrica da armadura (Ω)
5,65
5,68
3,78
3,67
3,82
3,84
3,73

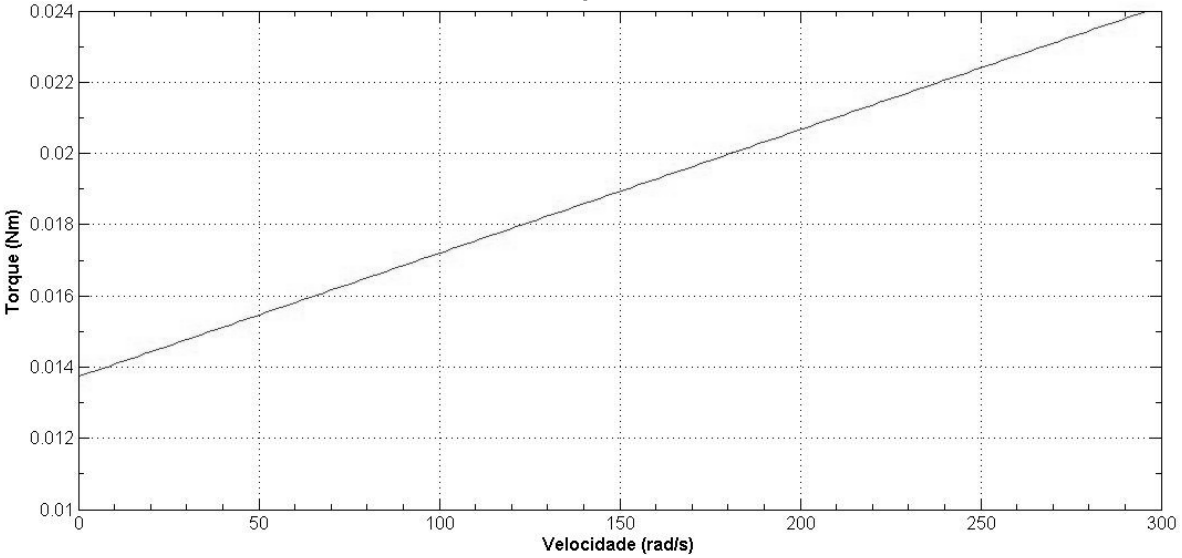
Em seguida, foi realizado o ensaio para a medição da constante K_E, que coincide com o valor de K_T. Nesse caso, foram escolhidos três valores de tensão que foram aplicadas ao motor. Utilizando a equação 10 e a resistência elétrica medida para determinar o valor de K_E, foi montada a Tabela 5.

Tabela 5 - Valores para o cálculo da constante K_E

Resistência (Ω)	Tensão (V)	Corrente (A)	ω (rad/s)	Ke (Vs/rad)
3,73	12,00	0,80	300,63	0,0299
	7,1	0,66	156,30	0,0296
	4,2	0,54	73,75	0,0296

Para o cálculo dos coeficientes de atrito é necessário construir o gráfico de “Torque x Velocidade”. O torque foi calculado a partir da constante K_E, como mostrado na equação 13 e foram utilizadas as medições da Tabela 5 para a construção do Gráfico 1.

Gráfico 1 - Torque x Velocidade



Utilizando o *software Matlab*® é possível extrair do gráfico os valores de B e F, como mostrado na equação 14. Os valores encontrados são:

- B = 3,47 x 10^{-5} Nms/rad
- F = 0,01373 Nm

O último parâmetro que deve ser obtido é o momento de inércia. Para isso, o motor foi ligado em sua tensão nominal e o sinal de velocidade, proveniente de um tacogerador, foi lido em um osciloscópio. A aquisição dos valores de tempo foi feita na própria tela do osciloscópio, como pode ser visto na Figura 37. O tempo “tb” tal que ω’ = 0.386 ωO é de aproximadamente 400ms. Portanto, aplicando a equação 14, o valor do momento de inércia do motor é de 1,39 x 10^{-5} Nms^2/rad.

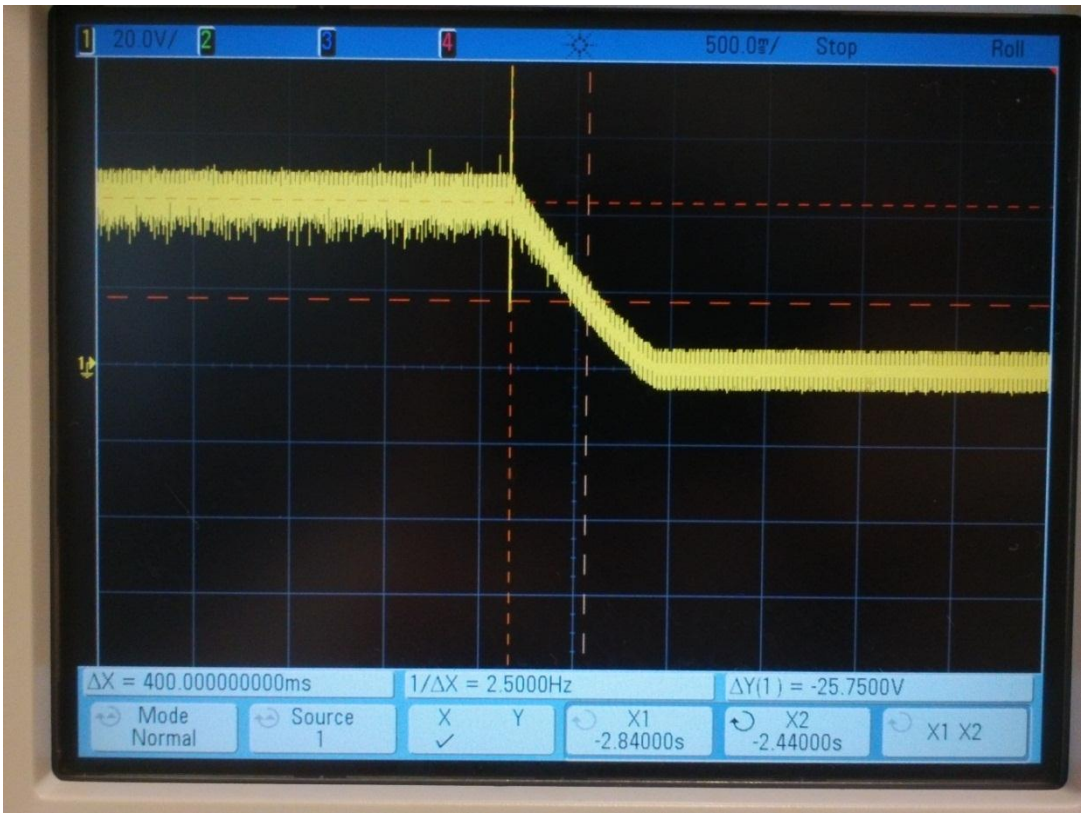
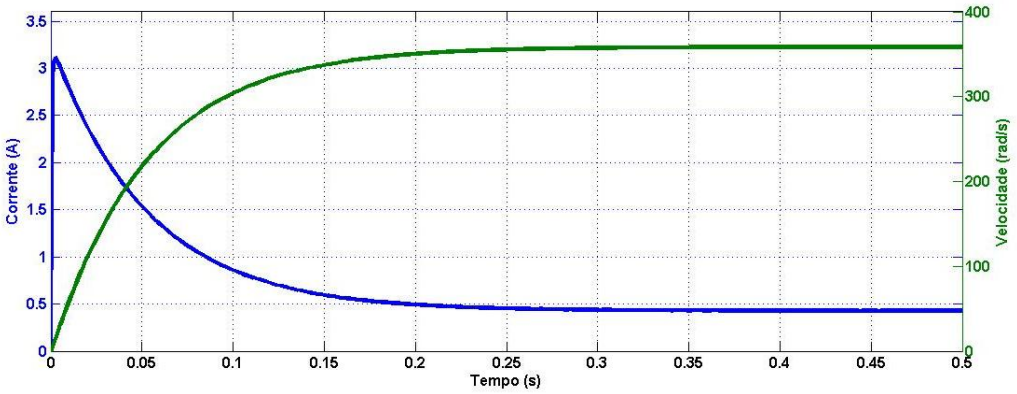


Figura 37- Velocidade do motor x Tempo

3.2 Simulação do controle do motor pelo modelo de função de transferência

Com os dados do motor já levantados, é possível iniciar uma primeira análise da resposta do motor aos estímulos de entrada. Para isso, foi utilizado o primeiro modelo de motor CC. Aplicando uma entrada de 12V ao modelo obtém-se, como resposta, a saída de corrente e velocidade mostrada Gráfico 2. Esses dados são condizentes com o motor real, sendo que é possível a comparação com a Tabela 5. No caso da simulação, foi obtida a velocidade, após o transitório da partida, de 350 rad/s, sendo que a velocidade real do motor é de aproximadamente 300 rad/s. Já a corrente teve seu pico em 3,1A, e, no regime permanente, permaneceu em torno de 0,5A.

Gráfico 2 - Corrente e Velocidade do motor (F.T.) em malha aberta



Portanto, tendo um modelo adequado, pode-se iniciar o ajuste da malha de corrente do motor. Utilizando a ferramenta de auto-ajuste, foi observado que somente o ganho integrativo era necessário para se obter a resposta como mostrada no Gráfico 3. No Gráfico 4, observa-se a curva de corrente e velocidade pelo tempo. Percebe-se que a corrente se ajusta à referência de 0,3A rapidamente e sem apresentar sobressinal. Assim, é possível partir para a otimização da malha de velocidade mais externa ao modelo.

Gráfico 3 - Otimização da malha de corrente

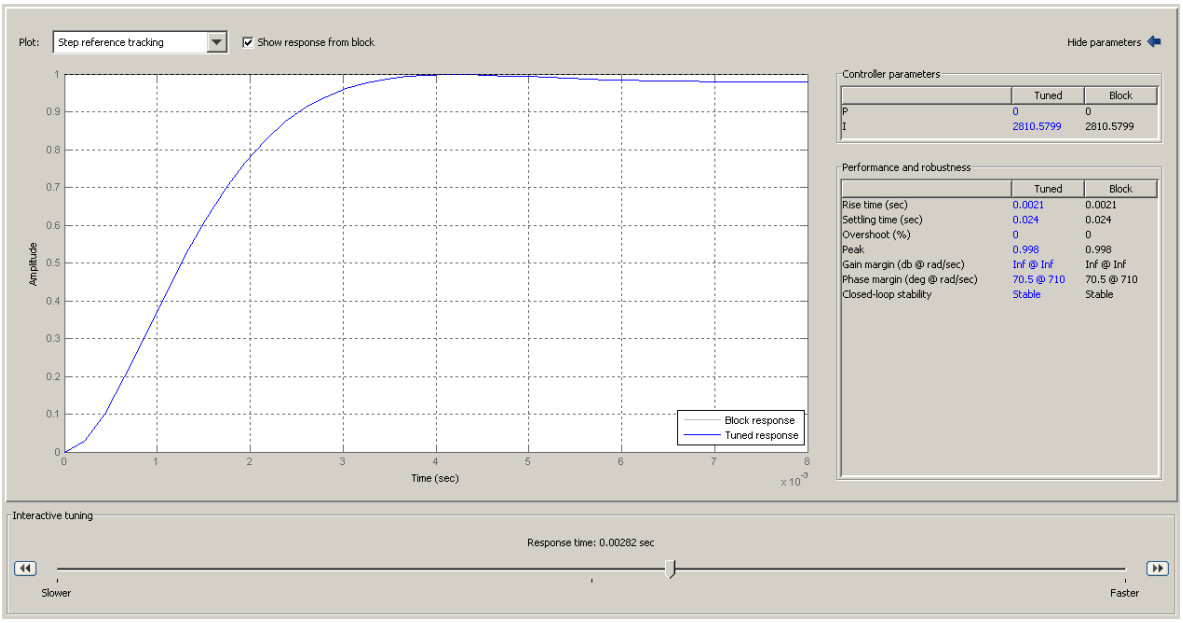
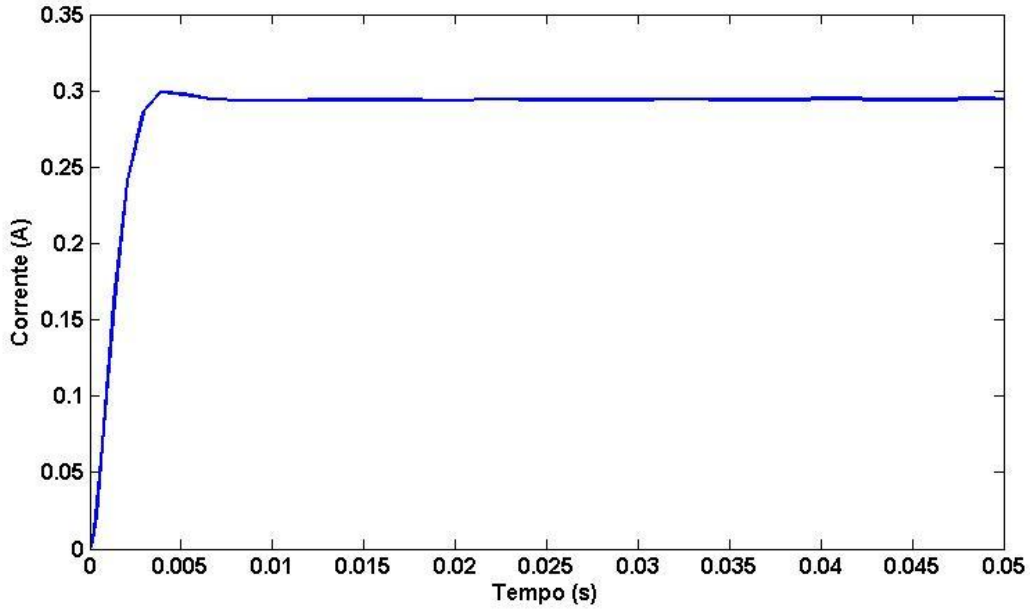


Gráfico 4 - Resposta do motor para referência de corrente em 0,3A



Adicionando-se a malha de velocidade, o auto-ajuste deverá realizar uma nova linearização da planta, com o propósito de calcular os ganhos PI dessa malha. É preciso

ter cuidado em relação à limitação da ação de controle na saída dos blocos, sendo que é necessário adicionar a saturação na saída. A ação de controle ajustada está mostrada no Gráfico 5 e a resposta da planta à velocidade de referência de 250 rad/s, no Gráfico 6.

Gráfico 5 - Otimização da malha de velocidade

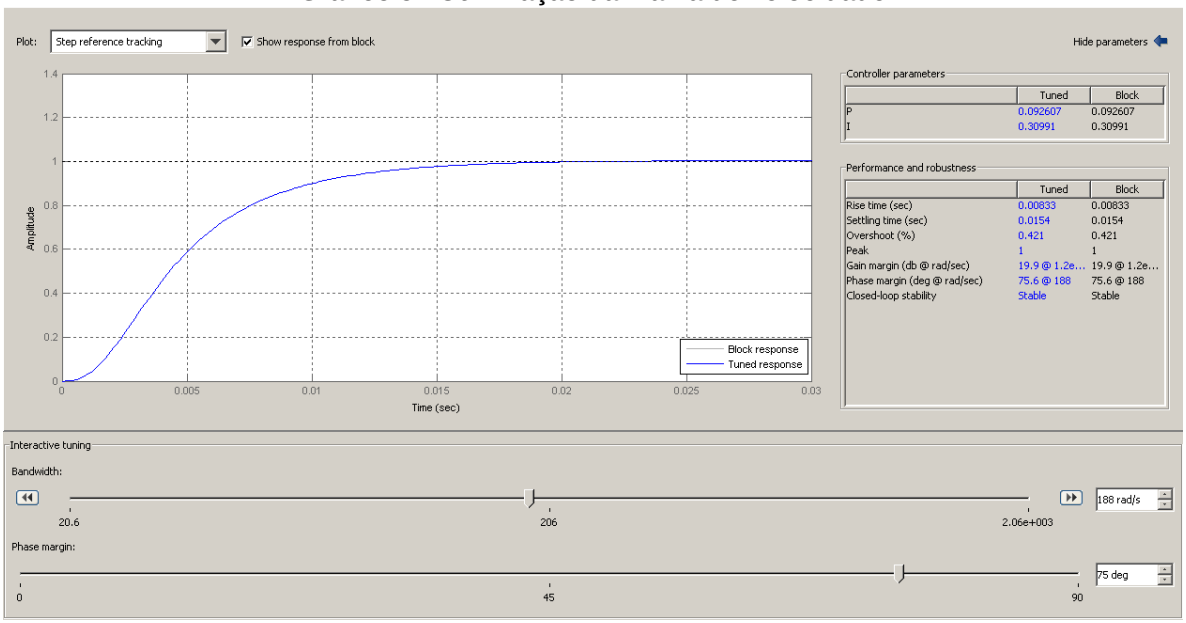
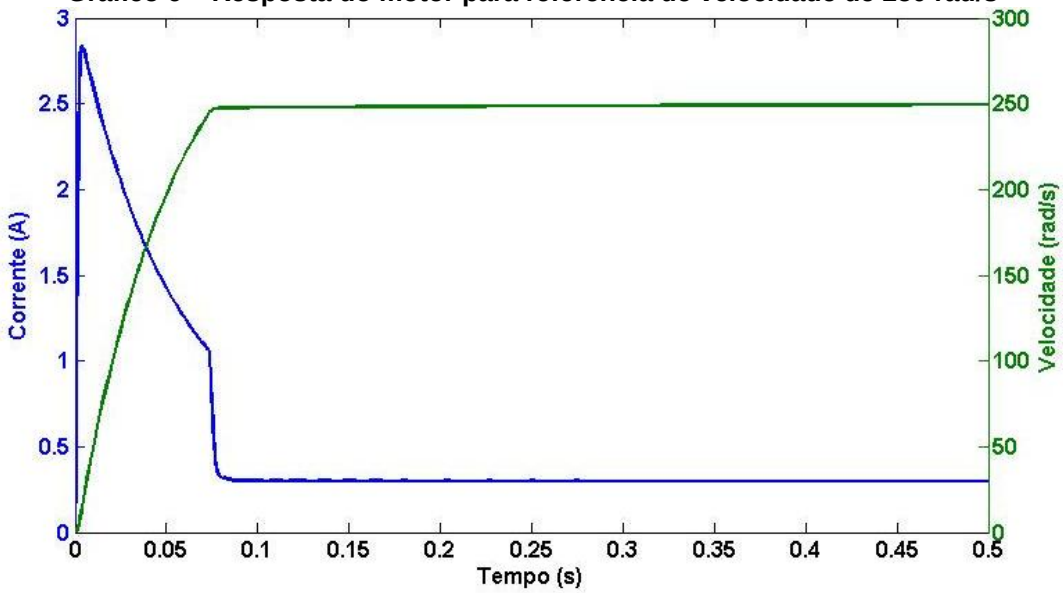


Gráfico 6 – Resposta do motor para referência de velocidade de 250 rad/s



Para o último passo da otimização das malhas, é necessário trabalhar nos ganhos de posição. A resposta obtida pela ferramenta de auto-ajuste foi satisfatória, apresentando um pequeno sobressinal (1,93%) para um tempo de resposta de acomodação de 0,333 segundos, como pode ser visto no Gráfico 7. A resposta de posição do motor para uma

volta (π rad) ocorreu em menos de 0,1 segundo e a corrente permaneceu em níveis aceitáveis, sem exceder o limite proposto de 3A.

Gráfico 7 - Otimização da malha de posição

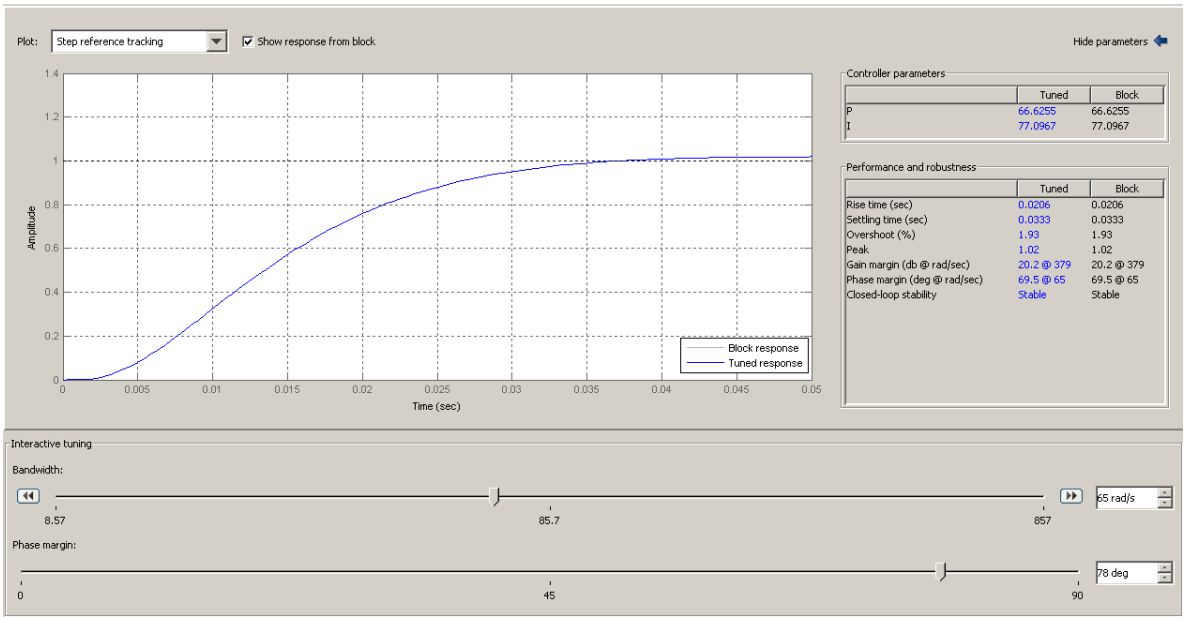
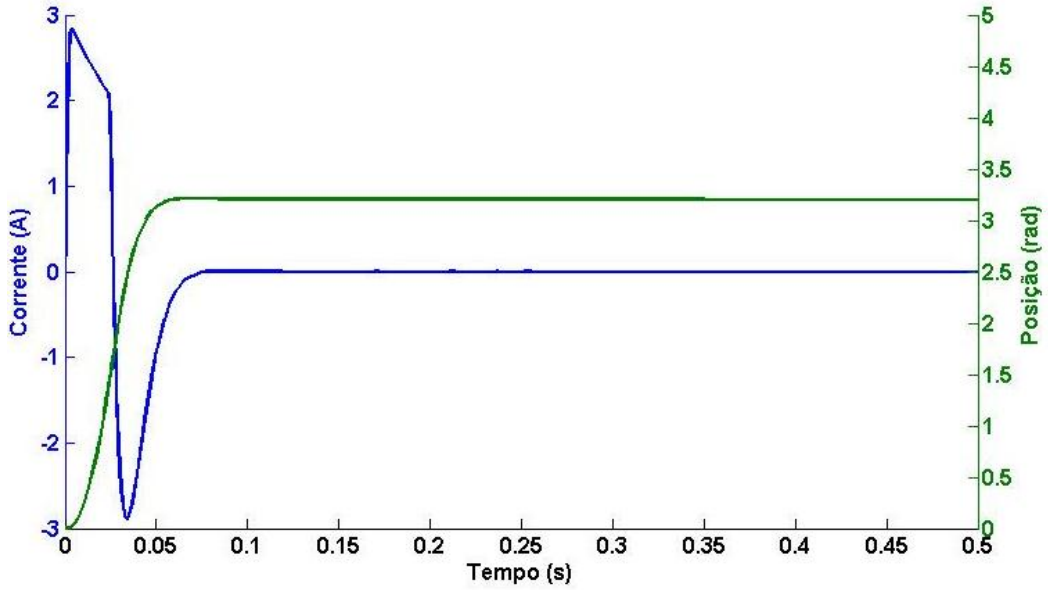


Gráfico 8– Resposta do motor para referência de posição π



3.3 Simulação do controle do motor no modelo com acionamento

Após o cálculo dos ganhos pelo modelo de motor por função de transferência, serão apresentados os resultados do controle ao incorporar-se o chaveamento à simulação.

Primeiramente, verifica-se a resposta do motor para o sistema em malha aberta e, nesse caso, o modelo se mostrou ainda mais próximo com relação aos resultados obtidos experimentalmente. O Gráfico 9 demonstra os resultados obtidos para uma entrada de 12V. Após estabelecer-se o regime permanente, a velocidade do motor se manteve em 300 rad/s e a corrente próxima a 0,8A.

Com o propósito de avaliar o controle obtido por meio da utilização do modelo por função de transferência, os valores de ganho obtidos anteriormente são inseridos diretamente no modelo com o circuito de chaveamento. A resposta do motor para uma referência de posição de π radianos pode ser vista no

Gráfico 10, onde é possível verificar que, para o modelo completo, o tempo de resposta do motor permanece igual ao modelo por função de transferência. Uma diferença claramente perceptível pode ser notada na corrente no motor, uma vez que apresenta maiores oscilações quando comparada ao modelo contínuo por função de transferência. Esse fato é devido ao chaveamento do circuito, que acrescenta uma oscilação à forma de onda da corrente.

Gráfico 9 - Corrente e velocidade do motor (modelagem completa) em malha aberta

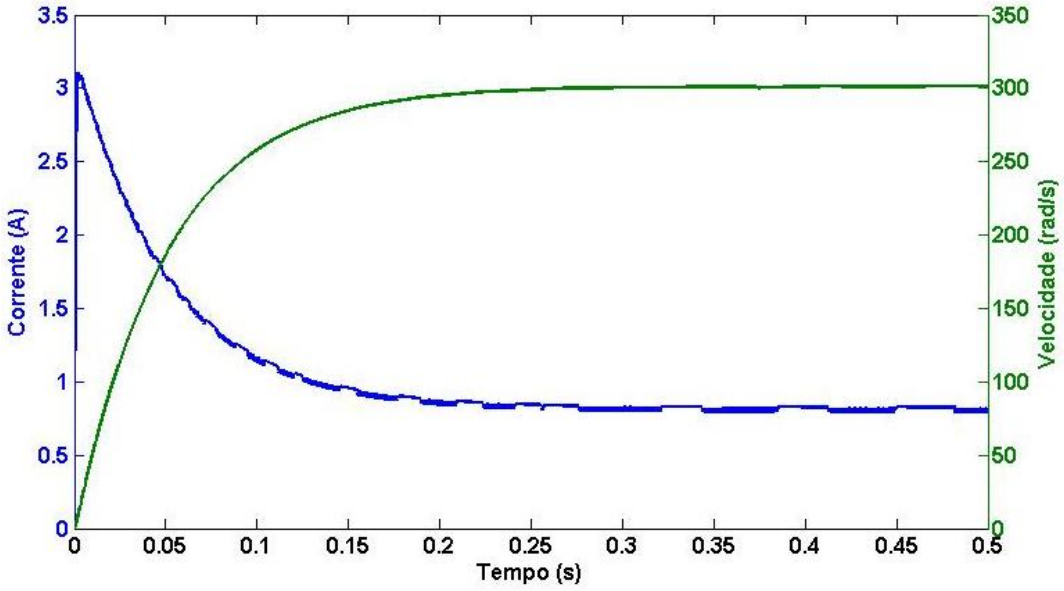
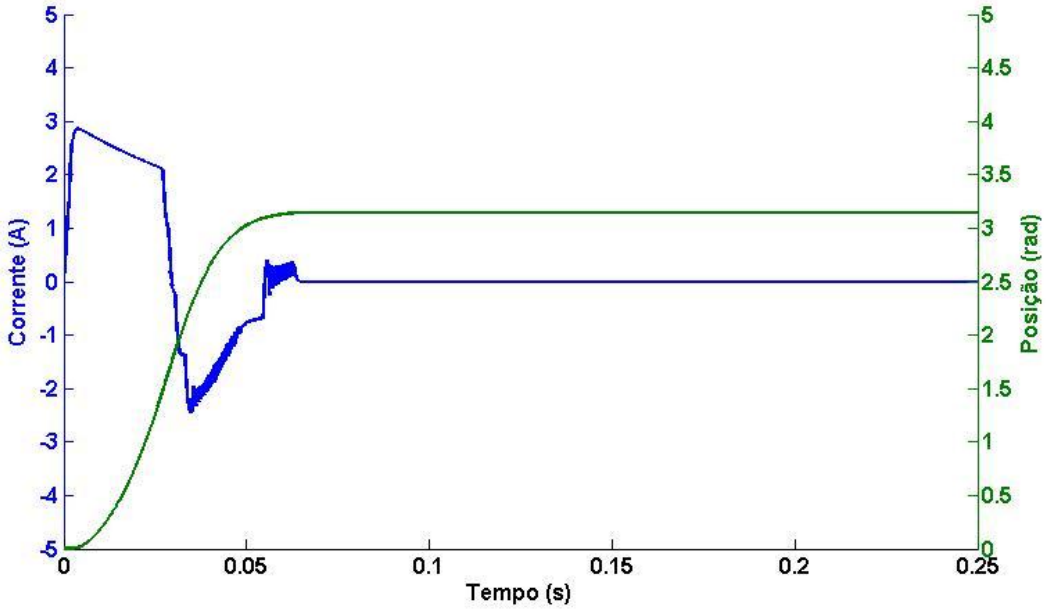


Gráfico 10 - Resposta do motor (modelo completo) para referência de posição π



3.4 Placas de potência e interface

O projeto dos circuitos de acionamento e de interface foi feito com o auxílio do *software Eagle*. Os sistemas já foram descritos separadamente e, após terem sido integrados, foram criados os esquemas mostrados na Figura 38 e na Figura 39, para a montagem das placas. Já as placas montadas são apresentadas na Figura 40 e na Figura 41.

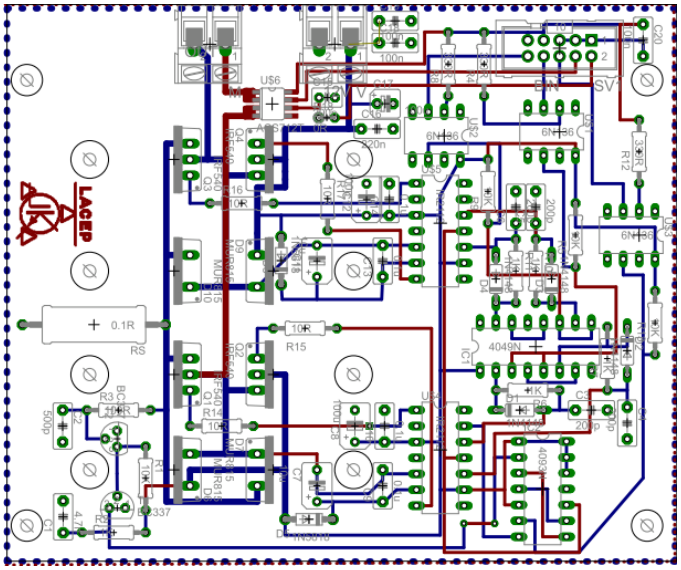


Figura 38 - Circuito de acionamento

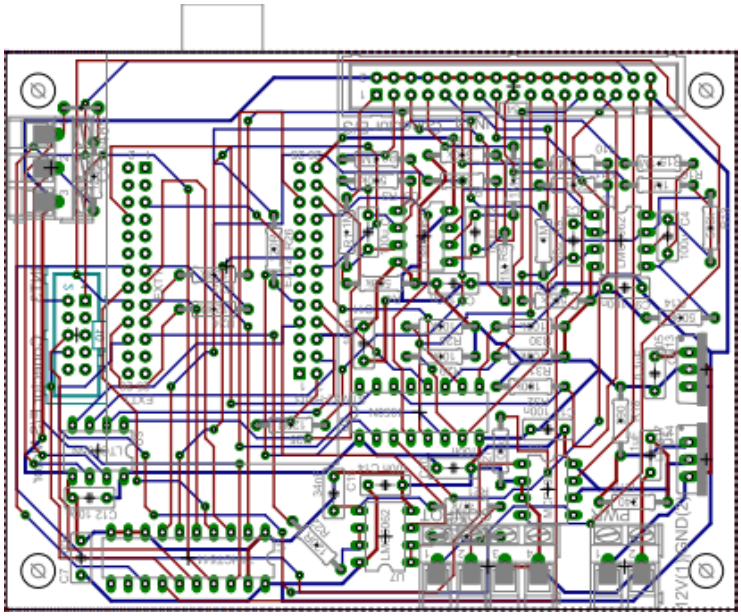


Figura 39 - Circuito de interface

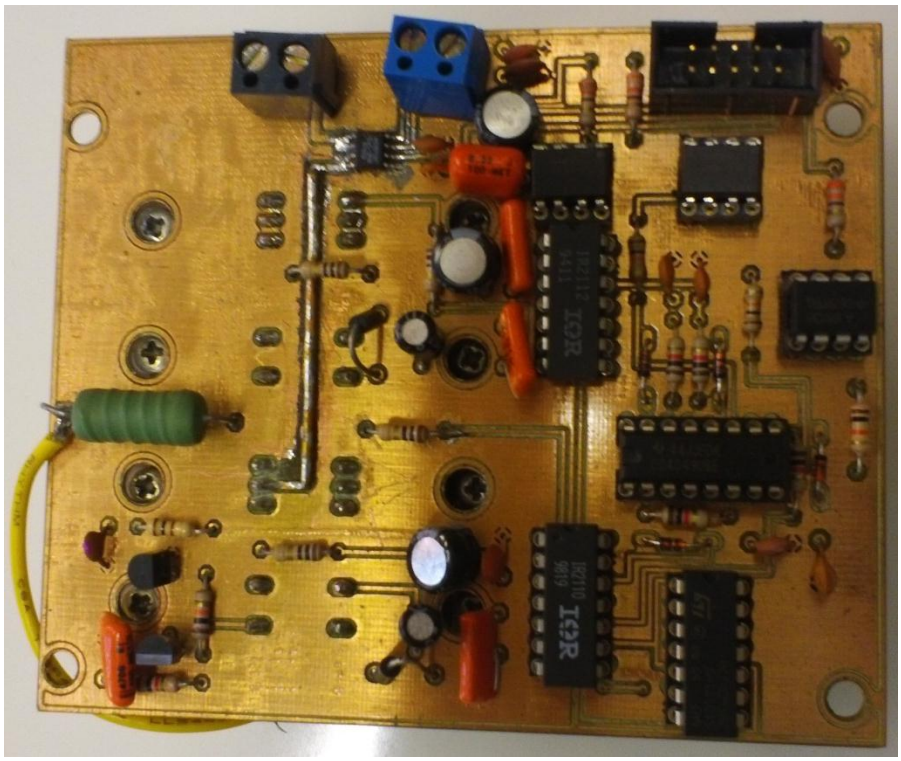


Figura 40 - Placa de acionamento completa

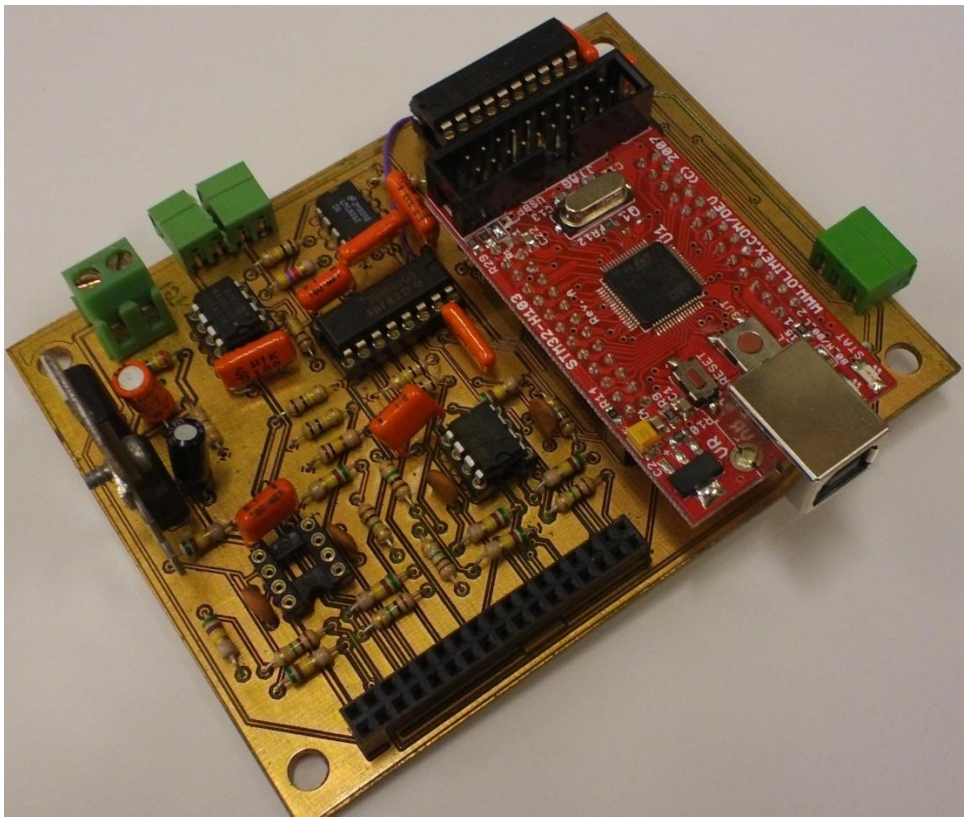


Figura 41 - Placa de interface completa

3.5 Controle do motor

Para o teste inicial do acionamento, foi criado um programa que aplica o PWM de acordo com a entrada desejada pelo programador. Foram realizadas medições para verificar o funcionamento do acionamento e adquiridos dados no osciloscópio.

Na Figura 42, o motor gira no sentido horário e tem-se a tensão aplicada ao motor, mostrada no sinal amarelo. O sinal roxo mostra o acionamento de um dos ramos da potência, acionando o motor no sentido horário.

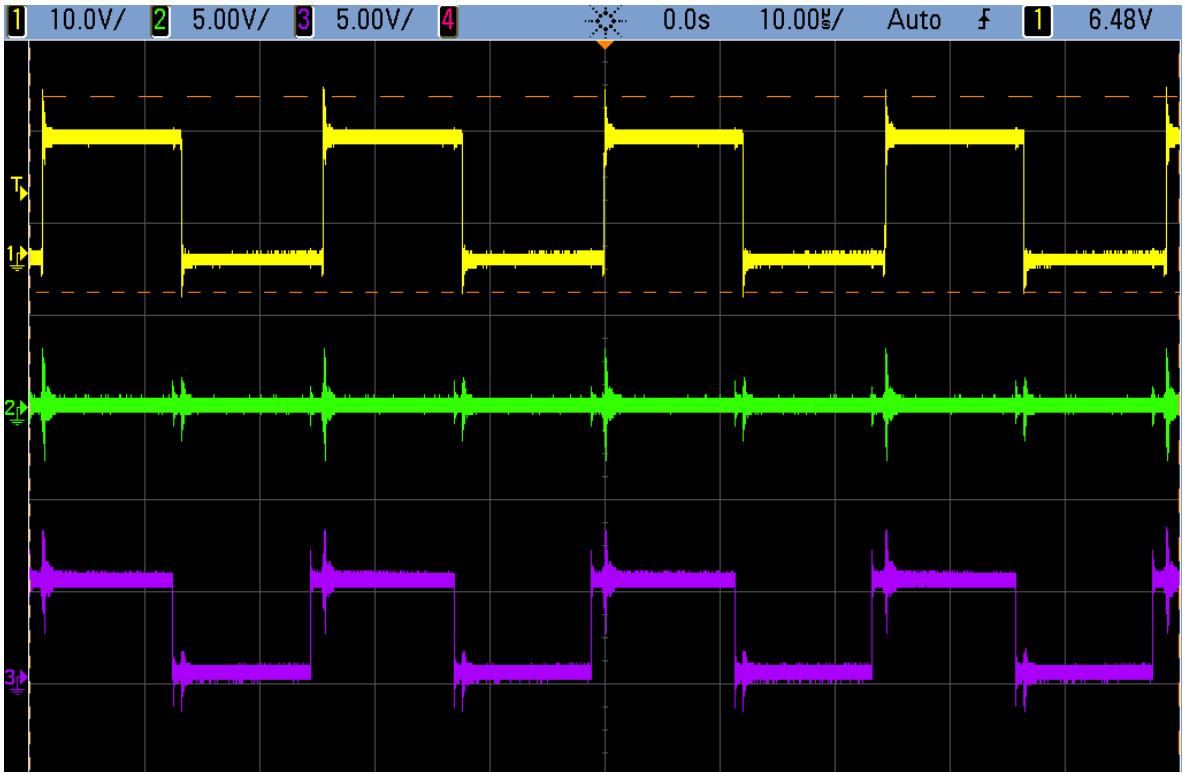


Figura 42 - PWM sentido horário

Ao comandar-se a inversão no sentido de rotação, o outro lado da ponte inversora é acionado. O PWM ligado agora é mostrado no sinal verde e a tensão sobre o motor está mostrada no sinal amarelo, como pode ser visto na Figura 43:

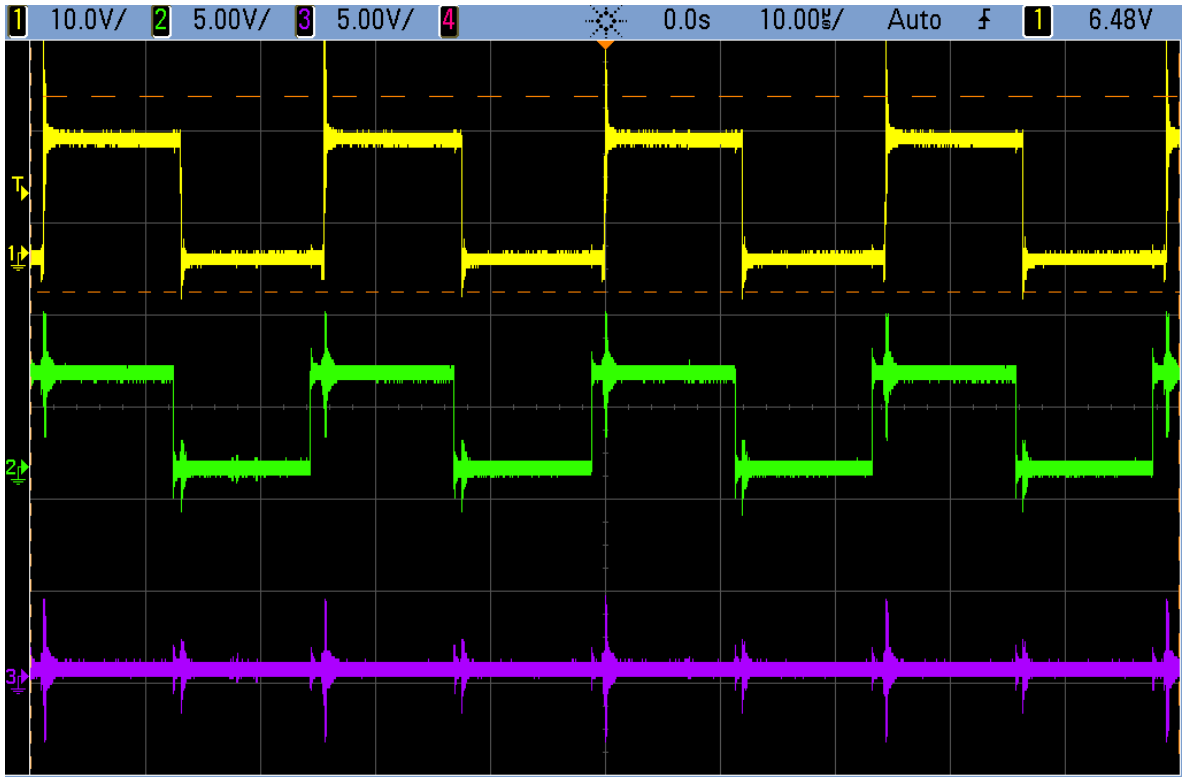


Figura 43 - PWM sentido anti-horário

Com a implementação do controle de posição, foram realizados testes para a sua regulação. A Figura 44 ilustra o movimento do motor buscando a posição de referência. O tempo necessário para o posicionamento de um valor de referência de 2 para 6 radianos (Δ de 4 radianos) no redutor, acoplado ao motor, com relação de 100:1 é de 1,34 segundos. Considerando uma variação no motor de 400 radianos, foi feita uma nova simulação com o intuito de comparar os resultados. No modelo simulado, a resposta do motor foi em um tempo de aproximadamente 1,4 segundos, como mostrado no Gráfico 11.

Após atingir a posição desejada, foi adicionado um torque externo para simular perturbações. A Figura 45 mostra o PWM enviado ao circuito de potência, com os sinais em roxo e verde. Já na Figura 46, adicionando-se um torque externo, é possível verificar que o período do PWM do sinal roxo é aumentado proporcionalmente, assim como a tensão de saída, uma vez que o motor busca manter a posição de referência, compensando também o torque externo.

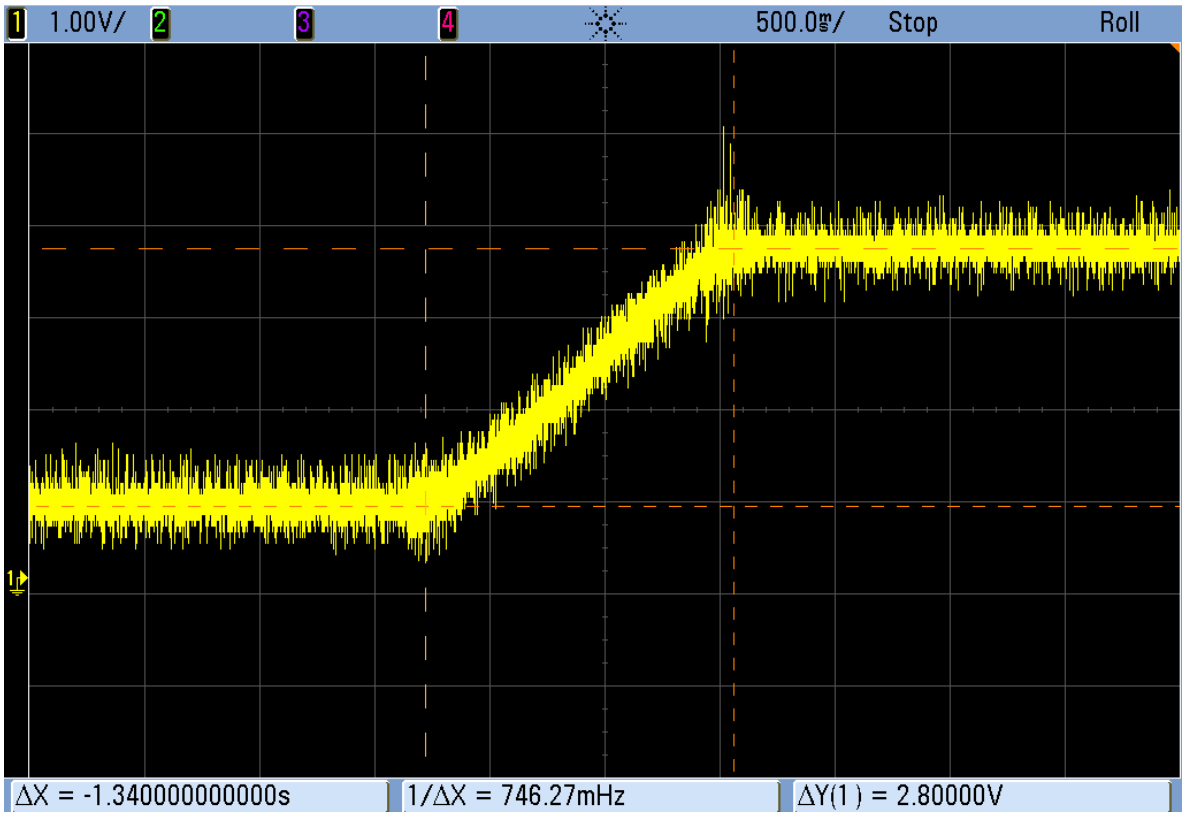
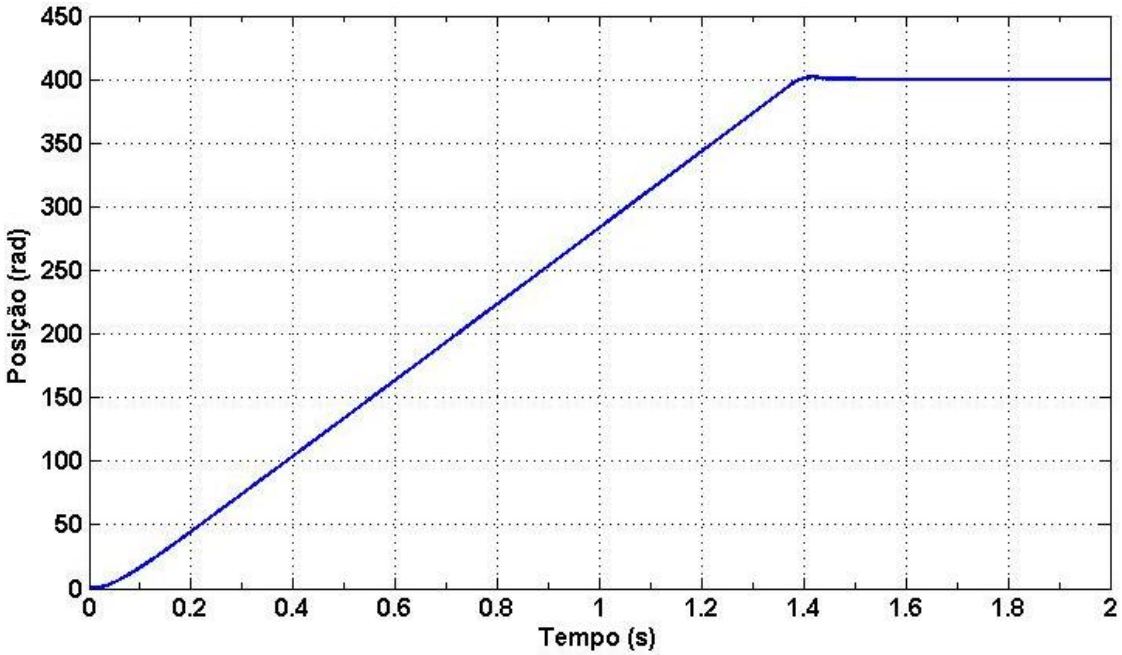


Figura 44 - Posicionamento com Δ de 4 radianos no redutor

Gráfico 11- Posicionamento no modelo com Δ 400 rad



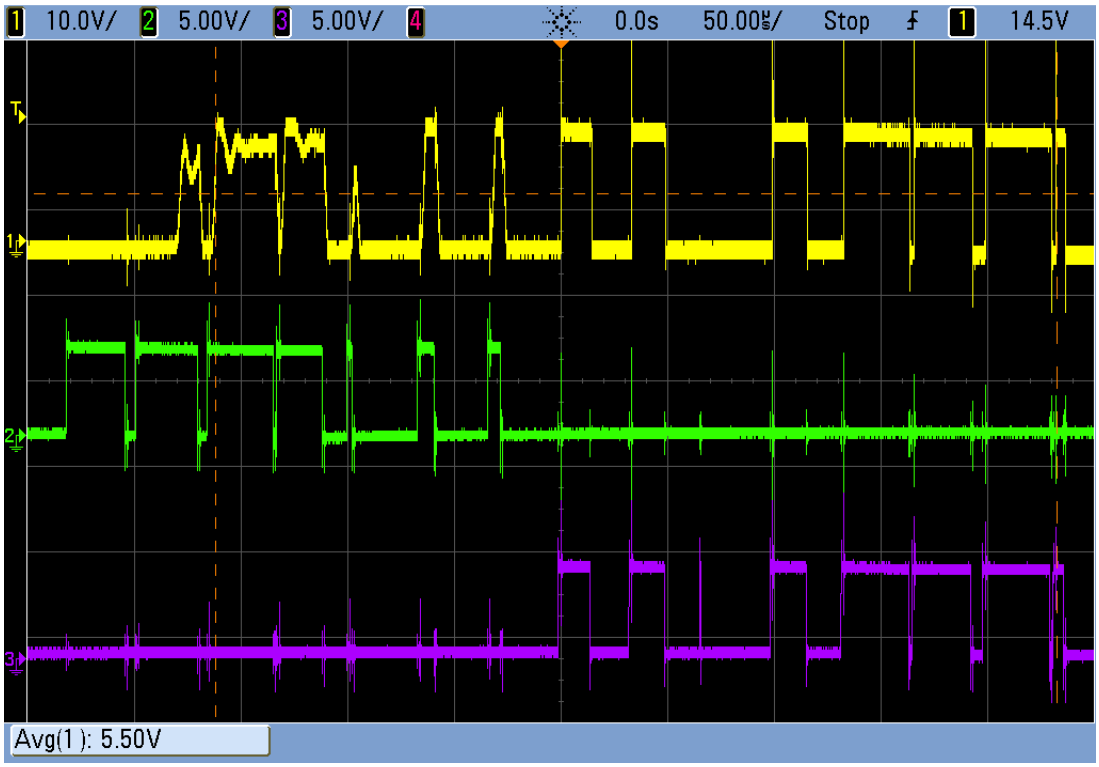


Figura 45 - Motor posicionado sem torque externo

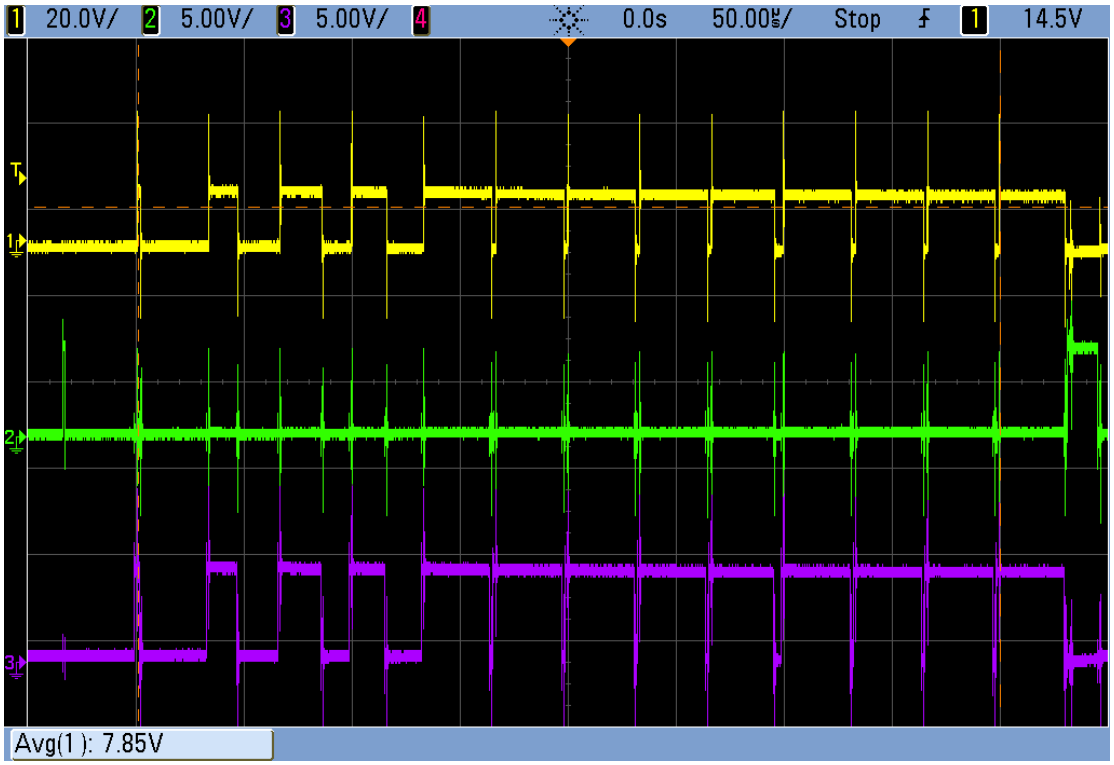


Figura 46 - Motor posicionado com torque externo

4. Conclusão

Para a construção de um controle de posição para motores, é necessária a integração dos atuadores mecânicos, dos sensores e também do controle eletrônico. Além dessa integração, o papel da simulação é importante para estudar previamente o comportamento do motor e, desse modo, calcular as ações necessárias para o seu controle.

A partir dos parâmetros do motor, é possível conseguir uma simulação condizente com a realidade e, portanto, o cálculo dos ganhos das malhas de corrente, velocidade e posição podem ser feitos a partir da modelagem digital.

O microprocessador ARM Cortex-M3 mostrou-se eficaz no gerenciamento do sistema de controle, apesar de não suportar, nativamente, variáveis em ponto flutuante. Contudo, esse detalhe pode ser contornado aplicando as técnicas descritas no trabalho, tais como a rotação de *bits* que são executadas com menos ciclos de processamento.

As vantagens da aplicação desse microprocessador ao controle de sistemas são o seu baixo custo (YIU, 2009), o desenvolvimento de *softwares* a partir de ferramentas gratuitas, as diversas interfaces que podem ser programadas e a sua frequência de trabalho em 72MHz.

Bibliografia

BRAGA, N. C. O que significa Rail-to-Rail **Saber Eletrônica**, São Paulo, v. 442, Novembro 2009.

BROWNELL, D. et al. OpenOCD User's Guide. **OpenOCD**, 18 Outubro 2011. Disponivel em: <http://openocd.sourceforge.net/doc/html/index.html>. Acesso em: 30 Outubro 2011.

FITZGERALD, A. E.; KINGSLEY JUNIOR, K.; STEPHEN, D. **Máquinas Elétricas**. 6ª Edição: Bookman, 2006.

HAUGEN, F. Styring av mekatroniske systemer. **Buskerud University College**, 2008. Disponivel em: <http://techteach.no/fag/sesm3401/h08/>. Acesso em: 23 Setembro 2011.

HERRMANN, U. Building an ARM cross-toolchain with binutils, gcc, newlib, and gdb from source, 03 Março 2009. Disponivel em: <http://www.hermann-uwe.de/blog/building-an-arm-cross-toolchain-with-binutils-gcc-newlib-and-gdb-from-source>. Acesso em: 30 Outubro 2011.

MESSNER, W.; TILBURY, D. Control Tutorials For Matlab and Simulink, 1998. Disponivel em: <http://www.engin.umich.edu/class/ctms/index.htm>. Acesso em: 8 Outubro 2011.

OGATA, K. **Engenharia de Controle Moderno**. 4ª ed. Upper Saddle River, Nova Jersey: Prentice Hall, 2003. 104-105 p.

OLIMEX. STM32-H103 HEADER BOARD FOR STM32F103RBT6 CORTEX-M3 MICROCONTROLLER. **Olimex Developmento boards and Tools**, 2008. Disponivel em: <http://www.olimex.com/dev/pdf/ARM/ST/STM32-H103.pdf>. Acesso em: 29 Outubro 2011.

OLIVEIRA, V. A.; AGUIAR, M. L.; VARGAS, J. B. **Sistemas de Controle - Aulas de Laboratório**. 3ª. ed. São Carlos: EESC-USP, 2005. ISBN 85-85205-49-0.

PAGAC, D.; NEBOT, E. M.; DURRANT-WHITE, H. An evidential aproach to map-building for autonomous vehicles. **IEEE transactions on robotics and automation**, Agosto 1998. Volume 14, Número 4. pp623-629.

PATANÉ, E. J. **Implementação de controle de velocidade em malha fechada para motores de corrente contínua utilizando sistema de aquisição de dados**. Dissertação de mestrado, 123p.,Escola de Engenharia Mauá. São Caetano do Sul. 2008.

RASHID, M. H. **Power Electronics**. 2a. ed. São Paulo: Makron Books Brasil, 2003.

SHILLER, Z.; GWO, Y.-R. Dynamic Motion Planning of Autonomous Vehicles. **IEEE transactions on robotics and automation**, 7, Abril 1991.Volume 7, número 2, 241-249.

STMICROELECTRONICS. STM32 - 32-bit ARM Cortex MCU. **STMicroelectronics**, 2011. Disponivel em: <http://www.st.com/internet/mcu/class/1734.jsp>. Acesso em: 31 Outubro 2011.

THAM, M. Discrete PID Controllers. **University of Newcastle Upon Tyne**, Newcastle, 1998. Disponivel em: <http://lorien.ncl.ac.uk/ming/digicont/digimath/dpid1.htm>. Acesso em: Outubro 20 2011.

YIU, J. **The Definitive guide to the ARM CORTEX-M3**. 2ª ed. Oxford : Newnes, 2009.

"**IEEE Standard Test Access Port and Boundary - Scan Architecture**," *IEEE Std 1149.1-1990*,1990 doi: 10.1109/IEEESTD.1990.114395.

Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=211226&isnumber=5509> Acesso em: 06 Novembro 2011.