

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE ENGENHARIA MECÂNICA
ENGENHARIA MECÂNICA

JOANA PEREIRA REPINALDO

**CONTROLE DE VELOCIDADE DE UM SERVOMOTOR UTILIZANDO
SOFTWARE LABVIEW REAL-TIME**

TRABALHO DE CONCLUSÃO DE CURSO

CORNELIO PROCOPIO
2015

JOANA PEREIRA REPINALDO

**CONTROLE DE VELOCIDADE DE UM SERVOMOTOR UTILIZANDO
SOFTWARE LABVIEW REAL-TIME**

Trabalho de Conclusão de Curso de Graduação, apresentado ao Curso de Engenharia Mecânica, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Engenheiro

Orientador: Prof. Dr. Marcio Aurelio Furtado Montezuma.

CORNÉLIO PROCÓPIO
2015



Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio
Departamento Acadêmico de Mecânica
Curso de Engenharia Mecânica



FOLHA DE APROVAÇÃO

Joana Pereira Repinaldo

Controle de Velocidade de um Servomotor Utilizando Software LabVIEW Real-Time

Trabalho de conclusão de curso apresentado às 14:00hs do dia 13/11/2015 como requisito parcial para a obtenção do título de Engenheiro Mecânico no programa de Graduação em Engenharia Mecânica da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Avaliadora composta pelos professores abaixo assinados. Após deliberação, a Banca Avaliadora considerou o trabalho aprovado.

Prof(a). Dr(a). Marcio Aurelio Furtado Montezuma - Presidente (Orientador)

Prof(a). Me(a). José Aparecido Lopes Junior - (Membro)

Prof(a). Me(a). Conrado Di Raimo - (Membro)

AGRADECIMENTO(S)

Agradeço primeiramente a Deus, que ilumina meu caminho e renova a minha força a cada momento.

Ao Orientador deste trabalho, Prof. Dr. Marcio Aurelio Furtado Montezuma, por compartilhar seus conhecimentos, pela confiança, paciência e incentivo.

Aos meus avós, pelos valores e ensinamentos sem os quais jamais teria me tornado quem sou.

Aos meus pais, Maria José e Francisco, por acreditarem sempre no meu potencial e não medirem esforços para realização dos meus sonhos.

Ao meu pai Fernando, que infelizmente não está presente neste momento, mas sempre foi meu exemplo de amor, luta e superação.

Ao meu irmão Vitor, pelo carinho e por toda a ajuda.

Ao meu namorado Fábio, pela compreensão, dedicação e amor, e por estar ao meu lado em todos os momentos.

Aos meus amigos, pelos momentos de descontração e pelo apoio.

Aos companheiros do laboratório LaSisC, pelos conhecimentos e experiências compartilhadas, as quais foram fundamentais para a realização deste trabalho.

A todos os professores da UTFPR, campus Cornélio Procópio, que foram tão importantes na minha vida acadêmica.

A PROGRAD - Pró-Reitoria de Graduação e Educação Profissional da UTFPR - pelo suporte financeiro.

Meus agradecimentos a todos que de alguma forma contribuíram para que a conclusão deste trabalho se tornasse possível.

RESUMO

REPINALDO, Joana Pereira. **Controle de Velocidade de um Servomotor Utilizando Software LabVIEW Real-Time**. 2015. 80 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia Mecânica. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

Este trabalho apresenta uma ferramenta didática para sistemas de controle que possibilita a associação entre os conceitos teóricos e práticos. Facilita o aprendizado ao proporcionar o desenvolvimento de experiências de laboratório. É possível aplicar técnicas de controle clássico como de controladores PID (Proporcional, Integral e Derivativo) permitindo ao usuário controlar a velocidade do servomotor. A entrada do sistema pode ser definida como uma forma de onda: quadrada, triangular ou senoidal. O controle do servomotor é realizado através de uma placa de aquisição de dados e do *software* LabVIEW, o qual possui uma interface gráfica amigável facilitando o manuseio do aluno. O sistema utiliza a técnica *hardware-in-the-loop* em tempo real e é capaz de armazenar os dados para análises qualitativas e quantitativas *off-line*.

Palavras-chave: Instrumentação. Sistema de Controle. LabVIEW. Controle em Tempo Real.

ABSTRACT

REPINALDO, Joana Pereira. **Speed Control of a Servo Motor Using LabVIEW Real-Time Software**. 2015. 80 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia Mecânica. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

This work presents an educational tool for control systems that enables the association between theoretical and practical concepts. It facilitates the learning by providing the development of laboratory experiments. It is possible to apply classic control techniques such as PID controllers (Proportional, Integral and Derivative) allowing the user to control servo motor speed. The system entry may be defined as a waveform: square, triangular or sinusoidal. The servo control is performed by a data acquisition card and the LabVIEW software, which has a friendly graphical interface that facilitates the handling by the student. The system uses the real-time hardware-in-the-loop technique and is capable of storing data for qualitative and quantitative offline analysis.

Palavras-chave: Instrumentation. Control system. LabVIEW. Real-Time Control.

LISTA DE ILUSTRAÇÕES

Figura 1 - Soma e Multiplicação independentes em LabVIEW	16
Figura 2 - Soma e Multiplicação dependentes em LabVIEW	16
Figura 3 - Painel Frontal em LabVIEW	17
Figura 4 - Diagrama de bloco em LabVIEW	18
Figura 5 - Exemplo Loops determinísticos e não determinísticos	21
Figura 6 - Comunicação utilizando protocolo de rede NI-PSP	23
Figura 7 - Funcionamento de Variáveis Compartilhadas Publicadas em Rede.....	24
Figura 8 - Funcionamento de Variáveis Compartilhadas Publicadas em Rede.....	25
Figura 9 - Bloco de fluxo de rede em LabVIEW	26
Figura 10 - Exemplo de comunicação de fluxo de rede	26
Figura 11 - Comunicação entre processos.....	27
Figura 12 - Bloco RT FIFO em LabVIEW	28
Figura 13 - Esquema de funcionamento RT FIFO.....	29
Figura 14 - Partes de um sistema de aquisição.	30
Figura 15 - Controlador em blocos	31
Figura 16 - (a) Sistema em malha aberta; (b) Sistema em malha fechada	32
Figura 17 - Diagrama de blocos controle proporcional.....	33
Figura 18 - Diagrama de blocos do controle proporcional e integral	35
Figura 19 - Diagrama de blocos do controle proporcional, integral e derivativo	36
Figura 20 - Bloco PID em LabVIEW	37
Figura 21 - Painel frontal do bloco PID em LabVIEW.....	37
Figura 22 - Encoder Optico	38
Figura 23 - Encoder Incremental	39
Figura 24 - Encoder Absoluto.....	40
Figura 25 - Codificação X1	40
Figura 26- Codificação X2.....	41
Figura 27 - Codificação X4.....	41

Figura 28 - Circuito de controle de corrente na carga	42
Figura 29 - Variação da largura de pulso	42
Figura 30 - Sistema completo.....	44
Figura 31 - Gerador acoplado ao servomotor.....	45
Figura 32 - (a) Pinos do driver de potência (b) Ligação dos pinos	46
Figura 33 - Placa de decodificação de quadratura	47
Figura 34 - Placa pull up elaborada no Eagle.....	47
Figura 35 - Ligação Placa pull up	48
Figura 36 - Interface para o usuário	49
Figura 37 - Variáveis compartilhadas publicada na rede.....	50
Figura 38 - Recebimento de dados através do fluxo de rede	50
Figura 39 - Armazenamento dos dados da aquisição	51
Figura 40 - Instalação da placa de aquisição	51
Figura 41 - Configuração dos canais de aquisição.....	52
Figura 42 - Programa executado no Target parte 1	53
Figura 43 - Programa executado no Target parte 2	54
Figura 44 - Programa executado no Target parte 3	55
Figura 45 - Envio de dados utilizando RT FIFO	55
Figura 46 - Envio de dados para o Host utilizando fluxo de rede	56
Figura 47 - Placa PCI-6221	57
Figura 48 - Pinos da placa de aquisição	57
Figura 49 - Ligação dos pinos da placa de aquisição.....	59
Figura 50 - Comunicação Ethernet.....	60
Figura 51 - Comunicação de dados do sistema	61
Figura 52 - Variação do ganho proporcional.	62
Figura 53 - Variação do ganho proporcional com $T_i = 0,001$	63
Figura 54 - Ação de controle para $K_p = 10$ e $T_i = 0,001$	63
Figura 55 - Ação de controle para $K_p = 1$ e $T_i = 0,001$	64
Figura 56 - Variação do ganho tempo integral com $K_p = 1$	64

Figura 57 - Variação do ganho tempo integral com $K_p = 0,9$	65
Figura 58 - Variação do ganho tempo derivativo com $K_p = 0,9$ e $T_i = 0,001$	65
Figura 59 - Comparação dos ganhos da função Autotunnig	66
Figura 60 - Resultado da velocidade do servomotor com inversão de rotação.	67
Figura 61- Resultado da variação da velocidade do servomotor.....	67
Figura 62 - Resultado da variação da velocidade com carga mecânica	68
Figura 63 - Resultado da variação dos tipos de onda.	68
Figura 64 - Resultado da velocidade com distúrbios no eixo do servomotor.....	69
Figura 65 - Diagrama de Blocos Host	76
Figura 66 - Diagrama de Blocos Target	77
Figura 67 - SubVI Somatória.....	78
Figura 68 - SubVI Velocidade.....	78
Figura 69 - SubVI Amostra	79
Figura 70 - SubVi Direção do motor	79
Figura 71 - SubVI Corrigir PWM.....	80

LISTA DE SÍMBOLOS

$u(t)$	Sinal na saída do controlador em função do tempo
K_p	Ganho proporcional
$e(t)$	Sinal de erro atuante em função do tempo
$r(t)$	Sinal de entrada em função do tempo
$y(t)$	Sinal de saída em função do tempo
$U(s)$	Sinal na saída do controlador no domínio complexo
$E(s)$	Sinal de erro atuante no domínio complexo
T_i	Ganho tempo integral
T_d	Ganho tempo derivativo
LP	Largura de pulso
T	Período
DC	Duty Cycle

LISTA DE ABREVIATURAS

PID	Proporcional, Integral e Derivativo
VI	Virtual Instrument
NI	National Instrument
PCI	Peripheral Component Interconnect
NI-PSP	National Instrument - Publish-Subscribe Protocol
TCP	Protocolo de controle de Transmissão
IP	Protocolo de internet
RT	Real-Time
URL	Uniform Resource Locator
FIFO	First-in-first-out
DAQ	Aquisição de dados
LED	Diodo emissor de luz
PWM	Pulse Width Modulation

SUMÁRIO

1 INTRODUÇÃO	12
1.1 DELIMITAÇÃO DO TEMA	13
1.2 OBJETIVOS	13
1.2.1 Geral.....	13
1.2.2 Específicos	13
1.3 ESTRUTURA DO TRABALHO	14
2 REVISÃO BIBLIOGRÁFICA	15
2.1 LABVIEW	15
2.1.1 INSTRUMENTO VIRTUAL	16
2.1.1.1 Painel Frontal	17
2.1.1.2 Diagrama de Bloco	18
2.1.2 LabVIEW Tempo Real.....	19
2.1.3 Comunicação de dados.....	21
2.1.3.1 Variáveis compartilhadas publicada na rede	22
2.1.3.2 Fluxo de Rede	24
2.1.3.3 RT FIFO	27
2.2 SISTEMA DE AQUISIÇÃO	29
2.3 SISTEMA DE CONTROLE	30
2.3.1 Controle Proporcional.....	32
2.3.2 Controle Proporcional e Integral.....	34
2.3.3 Controle Proporcional Integral e Derivativo	35
2.3.3.1 PID em LabVIEW	36
2.4 SERVOMOTOR.....	37
2.4.1 Encoder	38
2.4.2 Modulação de Largura de Pulso.....	41
3 METODOLOGIA E MATERIAIS	44
3.1 COMPONENTES DO KIT	44
3.1.1 Servomotor	45
3.1.2 Driver de potencia	45
3.1.3 Placa de decodificação em quadratura	46
3.2 SISTEMA DE AQUISIÇÃO E CONTROLE.....	48
3.2.1 Host	48
3.2.2 Target	51
3.2.3 Placa de aquisição	56
3.3 COMUNICAÇÃO DO SISTEMA	59
4 RESULTADOS	62
5 CONCLUSÃO	70
REFERÊNCIAS	71
APÊNDICES	76

1 INTRODUÇÃO

Atualmente as áreas de instrumentação e controle têm recebido grande destaque no setor industrial, pois as medições e o controle de todas as variáveis envolvidas nos processos de produção são de responsabilidade da mesma, sendo de grande importância e é parte fundamental dos processos industriais e de produção.

Estas áreas exercem um papel essencial no avanço da engenharia, pois atuam em sistemas importantes, como em sistemas de veículos espaciais, sistemas de direcionamento de mísseis, sistemas robóticos entre outros (NOGUEIRA, 2008).

Um das suas aplicações são em sistemas de controle de servomecanismo. Onde servomotores, motores acrescidos de sensores de movimento, são utilizados a fim de obter um controle eficiente da velocidade. Estes sistemas possuem uma variedade de aplicação e função, principalmente no setor da robótica (GOUVEIA; ANDRADE; BALTAZAR, 2011).

Um sistema de controle é uma combinação de componentes interligados de forma a manipular uma variável de entrada, com o objetivo de obter uma saída e um desempenho desejado do sistema, ou seja, um dispositivo que produz saída como resposta a uma entrada específica. O estudo desses sistemas se faz necessário para compreender e controlar as variáveis envolvidas em um processo, possibilitando o desenvolvimento de produtos confiáveis e econômicos para a sociedade (MELLO, 2015).

Para auxiliar e motivar os alunos no ensino de sistemas de controle, é importante a utilização de instrumentos didáticos, que não limitem o ensino apenas a aulas teóricas. Instrumentos como kits ou bancadas didáticas que possibilitam aplicar o conhecimento são capazes de demonstrar a relação entre os conceitos teóricos abordados em sala de aula com a sua aplicação prática na indústria, a fim de preparar o aluno para o mercado de trabalho (MARTINS; MELO, 2014).

Baseando-se nas considerações citadas acima, este trabalho consiste na construção de um kit didático para aquisição de dados e controle de velocidade utilizando a técnica de controle PID (Proporcional, Integral e Derivativo), como ferramenta de ensino e aprendizagem no curso de graduação em engenharia mecânica.

O sistema de aquisição e controle empregado, elaborado usando o ambiente de programação gráfica LabVIEW, combina uma interface intuitiva com a precisão do

sistema em tempo real. O LabVIEW foi utilizado, pois é uma linguagem de programação gráfica, que oferece amplo suporte para comunicação com instrumentos científicos e possui bibliotecas com um grande número de funções para diferentes campos. Ademais, também promove a visualização dos resultados do sistema através de gráficos, possibilitando uma visão do controle em tempo real conforme as variáveis são alteradas, o que facilita a compreensão e a comparação dos resultados práticos com os conceitos teóricos.

1.1 DELIMITAÇÃO DO TEMA

Este trabalho trata do projeto e implementação de um kit didático visando o desenvolvimento de um elemento de incentivo ao ensino. O sistema consiste na utilização de uma placa da National Instrument, para aquisição de dados e controle, do *software* LabVIEW e de um servomotor. Utiliza-se um VI (*Virtual Instrument*) para controle da velocidade com uma interface gráfica amigável, facilitando a análise e manipulação dos resultados e a gravação dos mesmos em formato de texto para análise posterior.

1.2 OBJETIVOS

Nesta seção são apresentados os objetivos gerais e específicos do trabalho apresentado.

1.2.1 Geral

Este trabalho tem como objetivo desenvolver um kit didático para utilização em experimentos dos conceitos aprendidos na disciplina de Controle e Servomecanismo.

1.2.2 Específicos

O sucesso do projeto depende da execução dos seguintes objetivos específicos:

- Desenvolver uma interface de controle do sistema para alteração dos parâmetros utilizados.
- Elaborar o programa para aquisição e controle da velocidade.
- Desenvolver circuitos eletrônicos do kit.
- Realizar testes e analisar resultados.
- Concluir e apresentar propostas para trabalhos futuros.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está dividido em 5 capítulos e terá a estrutura abaixo apresentada.

Capítulo 2 – Revisão Bibliográfica: será apresentado o estágio atual do estado da arte referente a este trabalho.

Capítulo 3 – Metodologia e Materiais: descrição das ferramentas e os métodos utilizados para o desenvolvimento do projeto.

Capítulo 4 – Análise dos Resultados: serão descritos os resultados experimentais obtidos e feitas as devidas análises.

Capítulo 5 – Conclusão: serão apresentadas as considerações finais sobre os resultados e sugestão de trabalhos futuros que possam ser implementados a partir do estudo realizado.

2 REVISÃO BIBLIOGRÁFICA

Para desenvolver um sistema de controle com aquisição de dados em tempo real é necessário compreender as partes por trás do funcionamento do sistema, tais como: o software LabVIEW e suas propriedades para aquisição e controle; a fundamentação teórica das técnicas de controle clássica; e o funcionamento de um servomecanismo. Assuntos que serão tratados no decorrer deste capítulo.

2.1 LABVIEW

O *software* de ambiente gráfico LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) desenvolvido pela National Instrument é utilizado por engenheiros e cientistas em sistemas de controle, através de representações gráficas que se assemelham a um fluxograma (NATIONAL INSTRUMENT, 2014). A programação é feita de forma simples, arrastando e soltando as funções desejadas, porém, o desenvolvimento do programa, dependendo da função utilizada, pode apresentar muitos detalhes e peculiaridades, os quais necessitam de um conhecimento mais aprofundado do programador.

Comparado a uma programação baseada em texto, a programação gráfica utilizada pelo LabVIEW encurta o tempo de desenvolvimento e melhora a legibilidade do código-fonte, pois disponibiliza bibliotecas internas com diversas funções (BOK; SCHAUER, 2011). Além de possuir suporte para comunicação com milhares de dispositivos de *hardware*, facilitando a comunicação com a placa de aquisição.

Outra diferença da programação gráfica em relação às baseadas em textos é o fato do código ser executado em paralelo em vez de forma sequencial. Na linguagem C, por exemplo, o fluxo dos dados são definidos linha por linha dentro de uma função, já no LabVIEW, assim que uma função recebe as entradas requeridas, ela começa a execução simultaneamente (ELLIOTT et al., 2007). Na figura 1, a soma e a multiplicação são independentes e executadas ao mesmo tempo, não importando a ordem que estão no programa.



Figura 1 - Soma e Multiplicação independentes em LabVIEW

Fonte: Autoria própria.

Caso seja necessária a utilização da forma de execução sequencial no ambiente gráfico, é indispensável o uso de estruturas que permitam a ligação dos dados. Um exemplo simples pode ser visto na figura 2, onde a soma é realizada e em seguida o resultado da soma é multiplicado, ou seja, uma das entradas da função de multiplicação não é válida enquanto a função soma não tenha terminado e passado os dados para a função multiplicação.

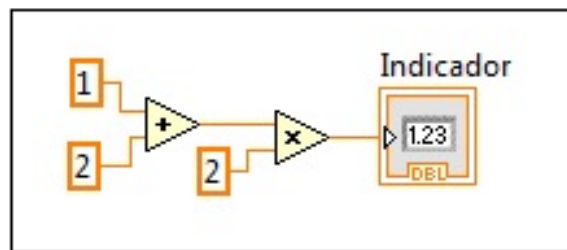


Figura 2 - Soma e Multiplicação dependentes em LabVIEW

Fonte: Autoria própria.

As rotinas de tarefas são armazenadas como um instrumento virtual (VI) que possui três componentes principais: o diagrama de blocos, onde é desenvolvida a programação, o painel frontal, onde se encontra a interface homem-máquina, e os conectores, os quais são os meios de conexão entre componentes de um VI (ELLIOTT et.al, 2007).

2.1.1 INSTRUMENTO VIRTUAL

Um instrumento virtual consiste na ligação entre um computador que possui um *software* que desenvolva a função necessária e um *hardware*, os quais, juntos, desempenham as funções dos instrumentos tradicionais, tais como osciloscópios, geradores de função, entre outros (HALVORSEN, 2011).

Os programas criados em LabVIEW recebem o nome de Instrumentos Virtuais, pois originalmente o *software* LabVIEW era destinado apenas a sistemas de controle utilizados no desenvolvimento de sistemas de instrumentação, onde as aplicações projetadas simulavam a funcionalidade dos instrumentos físicos (VIZCAINO; SEBASTIA, 2011).

O programa (VI) é constituído por um painel frontal e um diagrama de Blocos, os quais possuem funções específicas para criação do código e da interface.

2.1.1.1 Painel Frontal

O painel frontal é a interface com o usuário onde este pode controlar as variáveis do programa, seleccionando os valores das entradas e visualizando os dados de saída. Esta interação só ocorre quando o programa estiver em execução.

A construção do painel frontal é feita pelos controles e indicadores. Os controles simulam os comandos de um instrumento (botões e teclas), fornecendo dados de entrada ao diagrama de bloco, já os indicadores, por sua vez, simulam os mostradores de um instrumento, apresentando os dados de saída gerados no diagrama de blocos (gráficos ou outros) (COSTA, 2015).

O usuário é livre para criar seu painel da maneira que achar mais conveniente. A figura 3 mostra um exemplo de painel frontal, sendo (1) janela do painel frontal, (2) barra de ferramentas, (3) paleta de controles, (4) controles e (5) indicador.

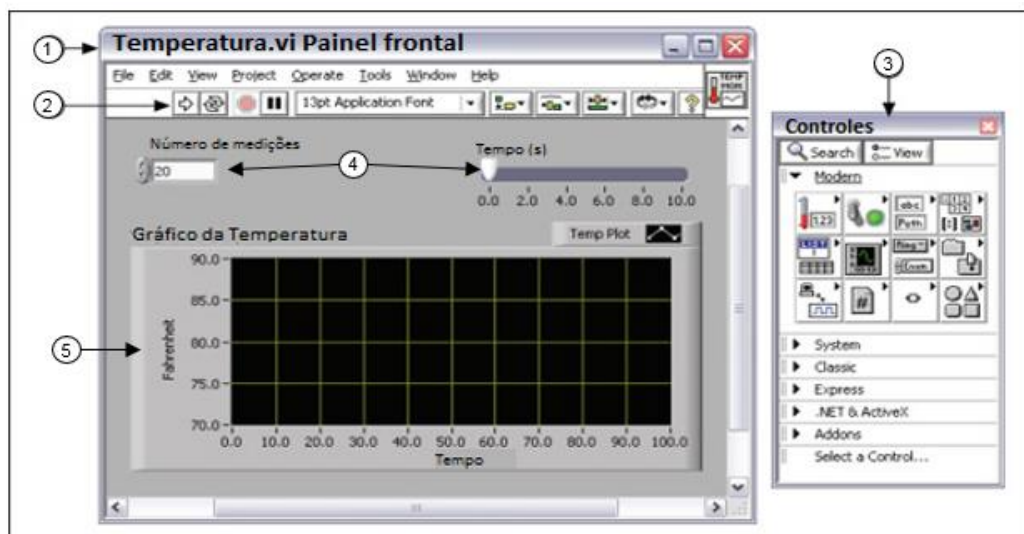


Figura 3 - Painel Frontal em LabVIEW

Fonte: Adaptado da National Instrument (2015).

2.1.1.2 Diagrama de Bloco

O diagrama de blocos contém o código fonte do programa através de representações gráficas das funções. Todos os objetos do painel frontal possuem um terminal no diagrama de blocos (COSTA, 2015).

Além dos terminais, do controle e indicadores, o diagrama de blocos possui funções, subVIs, constantes, estruturas e fios. Onde as funções são elementos de operação (soma, multiplicação, entres outros), os subVIs são VIs usados em outro VI, análogo a uma sub-rotina nas linguagens textuais. As estruturas são os elementos de controle do processo (*Loop For*) e os fios tem o papel de ligação dos dados (NATIONAL INSTRUMENT, 2015).

Na figura 4, o painel frontal, com seus controles e indicadores, é apresentado na janela esquerda, e o diagrama de bloco com os códigos desses controles e indicadores, além das funções desenvolvidas, na janela direita. Sendo (1) os terminais dos controles, (2) terminais de indicadores, (3) subVi, (4) função maior igual, (5) constante numérica e (6) estrutura (*Loop While*).

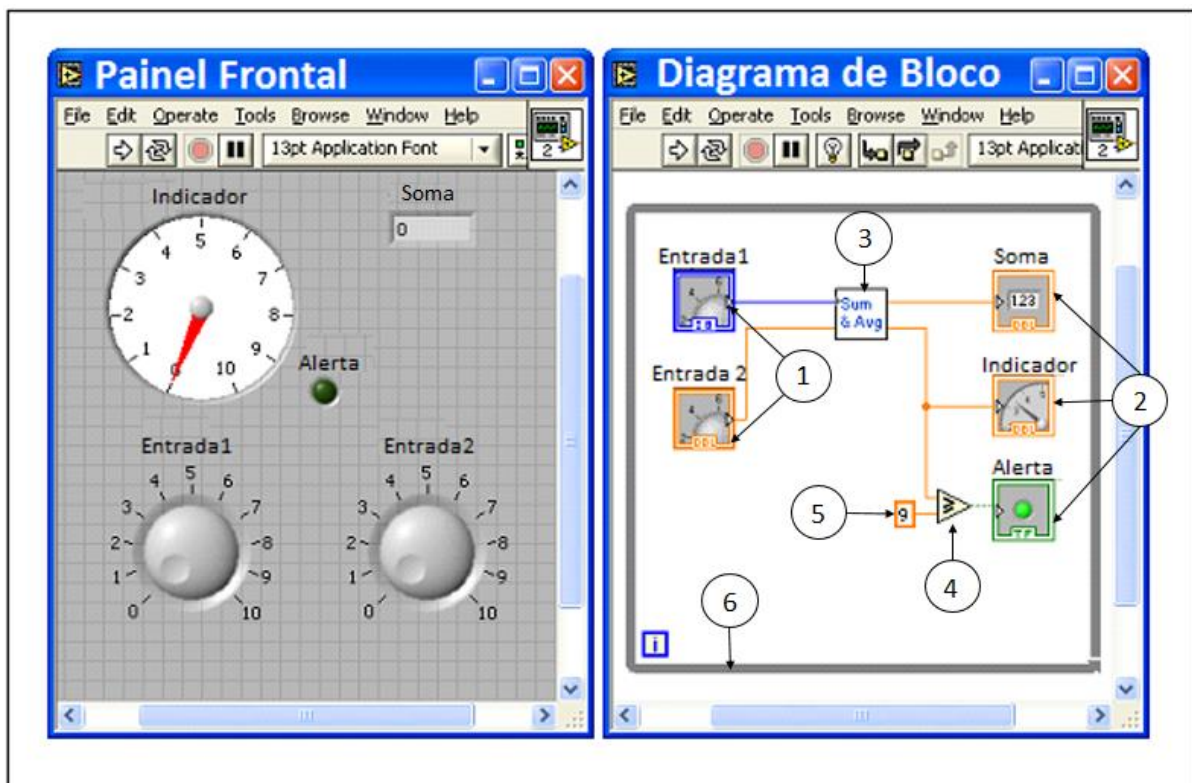


Figura 4 - Diagrama de bloco em LabVIEW

Fonte: Adaptado de Kehtarnavaz, Loizou e Rahman (2012).

2.1.2 LabVIEW Tempo Real

Sistemas em tempo real são sistemas que executam decisões em tempo crítico, ou seja, a execução de uma operação tem um tempo determinado para realização. Quando realizado no tempo estimado esta operação possui desempenho chamado de determinístico (NATIONAL INSTRUMENT, 2011).

Os sistemas em tempo real são empregados em operações de controle com o processador envolvido na malha fechada e a tomada de decisão feita em tempo crítico. Por exemplo, um equipamento coleta dados de entrada e envia estes dados diretamente para a memória, onde o sinal é analisado pelo processador e é enviado a resposta para o equipamento. Neste exemplo, o *software* está envolvido na malha garantindo uma resposta dentro de um espaço de tempo fixo.

Os sistemas operacionais de propósito geral estão na maioria dos computadores e executam simultaneamente várias aplicações, garantindo que todas tarefas recebam um tempo de processamento. Também respondem a interrupções de periféricos como mouse e teclado, interrompendo tarefas de alta prioridade por tarefas de mais baixa prioridade, não conseguindo prever um tempo de resposta para as aplicações críticas. Sendo assim, são utilizados no desenvolvimento e execução de aplicações não críticas, porém, não atendem aos requisitos para aplicações críticas. Alguns exemplos desses sistemas operacionais são o Microsoft Windows e o MAC OS, os quais podem atuar no controle e medição de tarefas não críticas, mas quando usados em aplicações determinísticas apresentam falhas (NATIONAL INSTRUMENT, 2011).

Ao utilizar um programa de controle, onde os dados devem ser executados em uma determinada taxa e sem interrupção, em um computador com sistema operacional de propósito geral como o Windows, isso será impossível, pois o sistema pode interromper a execução do controle para responder a alguma interrupção, ocasionando atraso no desempenho das tarefas (SEELAENDER, 2009).

Este tipo de programa deve ser utilizado em sistemas operacionais de tempo real, onde as tarefas podem ser priorizadas de modo que a tarefa mais crítica sempre será executada a tempo, garantindo confiabilidade ao sistema e previsibilidade do tempo (SEELAENDER, 2009).

Alguns acreditam que o sistema de tempo real é um sistema mais rápido que outros, porém, é apenas um sistema onde o tempo é previsto. Em sistemas de controle

para monitorar um sistema, o estado atual é comparado com o estado desejado e o tempo de execução desta malha é chamado de tempo de ciclo, o qual pode variar devido à complexidade do sistema (NATIONAL INSTRUMENT, 2011).

Um computador com sistema operacional de propósito geral pode ser convertido para um sistema em tempo real através do módulo LabVIEW Real-Time, disponibilizado pela NI (National Instruments), em conjunto com *hardware* dedicado, compatível com o computador. O *hardware* pode ser incorporado ao PCs Desktop através dos slots PCI (*Peripheral Component Interconnect*).

Para desenvolver uma aplicação em tempo real utilizando o LabVIEW, é importante identificar os processos não determinísticos dos determinísticos, pois os processos determinísticos necessitam concluir sua execução em um tempo determinado, precisando utilizar recursos de processadores dedicados para garantir isto. Ao separar as tarefas, garante-se que todas recebam os recursos suficientes para sua função. Por exemplo, em uma aplicação de aquisição de dados em intervalos definidos, com armazenamento dos seus dados em disco, a aquisição deve ser tratada de maneira determinista, enquanto o armazenamento dos dados é uma tarefa não determinística, pois esta operação possui tempos de resposta imprevisíveis que dependem de *hardware* (NATIONAL INSTRUMENT, 2014).

O LabVIEW fornece funções que permitem separar e programar estas tarefas dependendo de seu tempo de execução, com um alto grau de controle sobre as tarefas priorizadas e com capacidade de se certificar de que prazos são cumpridos. Garantindo que um programa seja executado com sincronismo. Isto é feito usando *Timed Loop* para definir a prioridade de uma tarefa, semelhante ao *While Loop* com recursos adicionais para executar de forma determinística, como especificar o período e a prioridade da execução do seu código. Quanto maior a prioridade do seu *Timed Loop* maior será a prioridade de execução de suas funções em relação às outras. Este *loop* é denominado de *loop* de tempo crítico (NATIONAL INSTRUMENT, 2014).

Na figura 5, tem-se o esquema resumido da aquisição de dados e armazenamento do exemplo anterior. Devido ao fato de a aquisição ser uma tarefa determinística, seu código está contido no *Timed Loop*, sendo executado com uma prioridade e tempo crítico maior que no armazenamento de dados contido no *While Loop*.

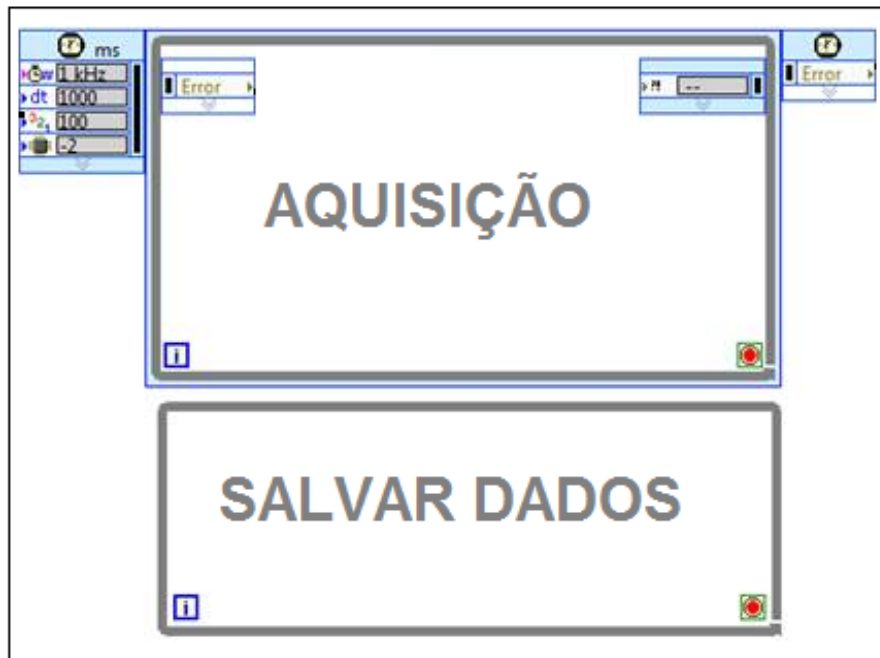


Figura 5 - Exemplo Loops determinísticos e não determinísticos
Fonte: Autoria própria.

Ao separar as tarefas em *loops* com estruturas diferentes, é de extrema importância a correta comunicação de dados para que nada seja perdido.

2.1.3 Comunicação de dados

O maior desafio do programador no processo de transmissão de dados no LabVIEW é a escolha correta da ferramenta de comunicação para sua aplicação. Existem diferentes tipos de comunicação entre processos que utiliza LabVIEW em tempo real. O site da NI fornece informações sobre os mecanismos de comunicação de dados e suas aplicações. Alguns métodos abordados neste trabalho são descritos no quadro 1 e detalhados posteriormente.

Método	Atributos
Variáveis compartilhadas publicada na rede (<i>Network-Published Shared Variable</i>)	Envia o valor mais recente.
Fluxo de Rede (<i>Network Streams</i>)	A transferência de alto rendimento de todos os pontos de dados.
RT FIFO	Usado como buffer para enviar dados de um local para outro sem perder qualquer valor.

Quadro 1 - Métodos de Comunicação de dados

Fonte: National Instrument (2014).

Neste trabalho será apresentado apenas a comunicação entre um computador que coleta os dados em tempo real e os envia através da porta Ethernet para ser executado com absoluta confiabilidade em outro computador, o qual lê e exibe os dados.

2.1.3.1 Variáveis compartilhadas publicada na rede

Network–Published Shared Variables, ou Variáveis Compartilhadas Publicadas em Rede, são ferramentas de compartilhamento de dados em LabVIEW. Sua implementação é feita de forma simples e abrange a maioria dos tipos de dados do LabVIEW. Possuem três partes principais: nós de variáveis de rede, variável compartilhada, e o NI *Publish-Subscribe Protocol* (NI-PSP). Os nós de variáveis de rede são funções colocadas no diagrama de blocos para ler e escrever operações da variável. A variável compartilhada é um espaço da memória que hospeda os dados publicados. Esta variável deve estar em execução em pelo menos uma das máquina em rede. NI-PSP é um protocolo de rede proprietário que otimiza o transporte das variáveis compartilhadas e é instalado no computador quando o *software* LabVIEW é instalado. É este protocolo baseado em TCP/IP que permite transmitir os dados através da rede (NATIONAL INSTRUMENT, 2014).

Protocolo de controle de Transmissão (TCP) e o protocolo de internet (IP) formam um protocolo de envio e recebimento de dados via internet. Este é uma espécie de linguagem comum utilizada na comunicação de dois computadores. Para

duas máquinas se comunicarem utilizando este protocolo, cada uma destas máquinas precisa ter um endereço IP diferente, pois é através do endereço IP que é possível identificar uma determinada máquina (BRANDINO, 1998).

O endereço IP é uma sequência de números composta de 32 *bits*, onde são separados em conjuntos de quatro sequências de 8 *bits*. Cada conjunto é separado por um ponto e recebe o nome de octeto, os dois primeiros octetos de um endereço IP podem ser utilizados para identificar a rede e os dois últimos octetos são usados na identificação de computadores (ALECRIM, 2011).

A figura 6 demonstra um exemplo da comunicação com Variáveis Compartilhadas Publicada em Rede e os endereços de IP para uma mesma rede.

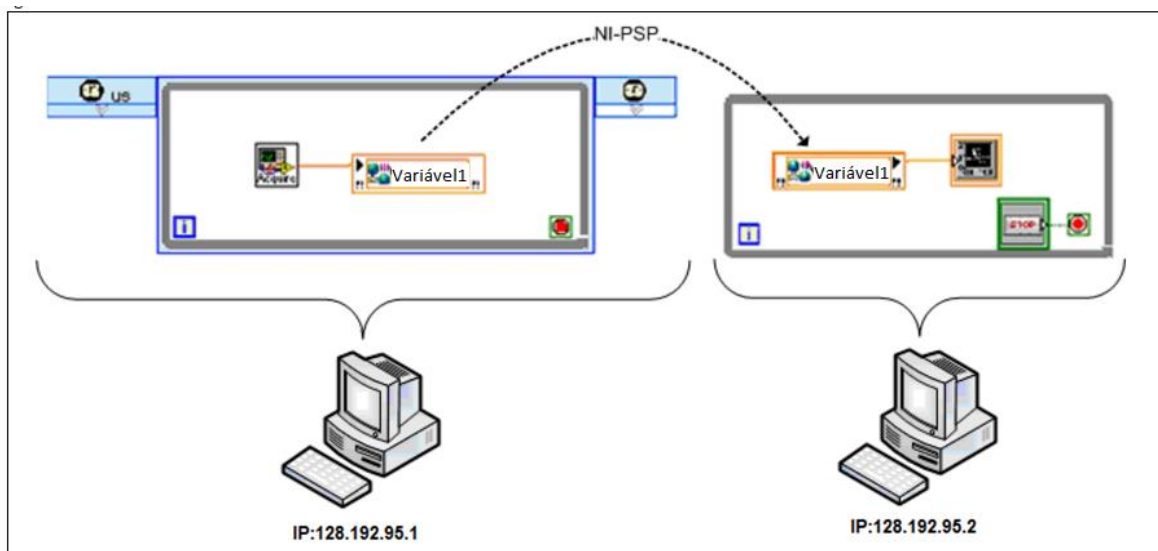


Figura 6 - Comunicação utilizando protocolo de rede NI-PSP

Fonte: Adaptado da National Instrument (2012).

A figura 7 apresenta um exemplo do funcionamento de Variáveis Compartilhadas Publicadas em Rede, onde, na aplicação A, o bloco da variável compartilhada (variável 1) recebe um valor. Este valor é enviado para um espaço na memória contida no computador 2, que atualiza seu valor e envia para as aplicações B e C. O aplicativo A não pode enviar seu valor diretamente para as outras aplicações, em vez disso envia para a memória, para que esta atualize todas as outras aplicações que contenham a variável, publicando somente valores mais recente.

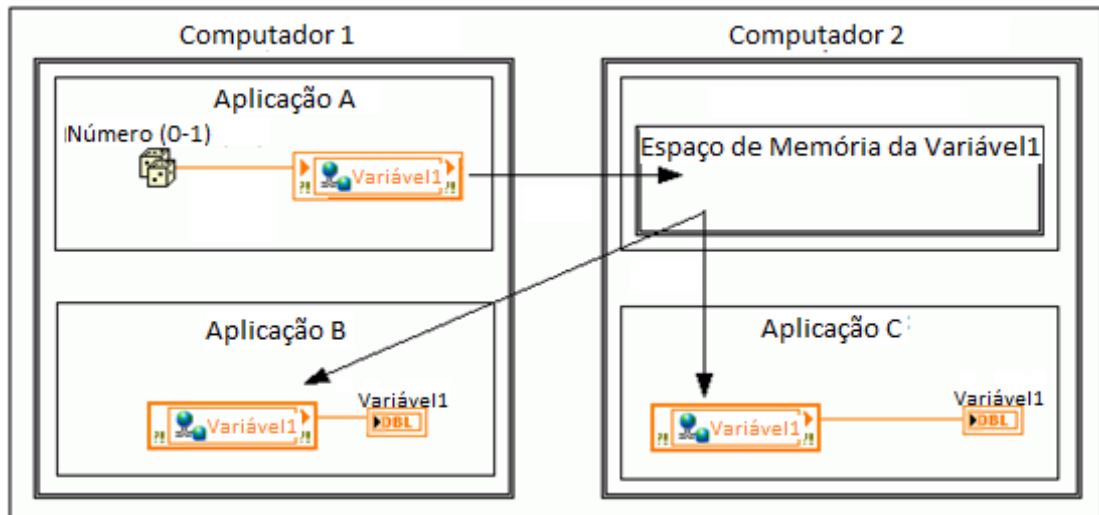


Figura 7 - Funcionamento de Variáveis Compartilhadas Publicadas em Rede

Fonte: National Instrument (2014).

Para prevenir a perda de dados devido a atrasos nas redes, existe a ativação de um *buffer*, mas este não garante a transferência de dados sem perdas. Por exemplo, se uma variável executa e grava dados de forma mais rápida que outra realiza a leitura dos dados, ocorrerá um estouro da memória e os dados serão perdidos. Mesmo se todas as variáveis executem a uma taxa rápida o suficiente, os dados ainda podem ser perdidos se a conexão TCP subjacente for interrompida devido a falhas na rede (COETZEE, 2015).

Esta técnica de transferência de dados é utilizada quando apenas o último valor é necessário. Quando a transferência ponto a ponto de dados é desejada, é necessária a utilização de Fluxos de rede.

2.1.3.2 Fluxo de Rede

Network Streams ou Fluxos de rede estão disponíveis a partir do LabVIEW 2010, os quais foram desenvolvidos para fornecer uma transmissão de dados eficiente. São usadas para transmitir dados de computador com sistema em tempo real para outro computador com sistema operacional de propósito geral. Utilizam taxas de transferência de dados que são comparáveis com TCP. No entanto, ao contrário do TCP, enviam os dados diretamente sem a necessidade de hospedar os dados em uma memória. O envio dos dados é unidirecional com canal de mão única de comunicação, que consiste de um escritor e um terminal leitor. Conforme a figura 8,

os dados são enviados em uma única direção. Por exemplo, o controlador RT (*Real-Time*) está lendo os valores da variável 1 enviados pelo cliente, e o cliente está lendo os valores das variáveis 2 e 3, enviado pelo controlador RT. Cada transferência tem um único fluxo com uma única direção (NATIONAL INSTRUMENT, 2013).

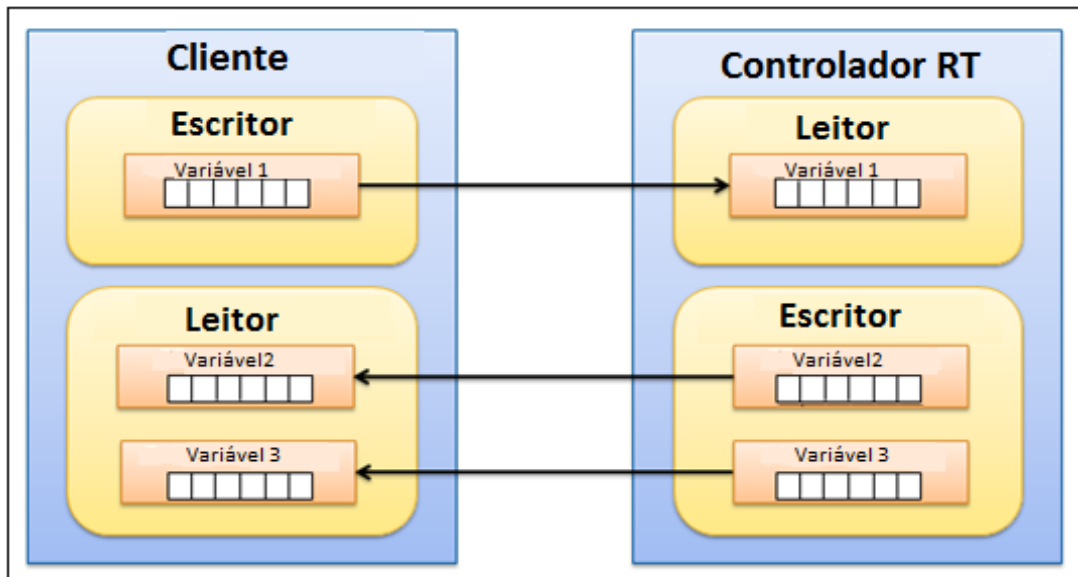


Figura 8 - Funcionamento de Variáveis Compartilhadas Publicadas em Rede

Fonte: Adaptado de Pollock (2014).

Em uma comunicação que utiliza fluxo de rede, um das extremidade é designada como a extremidade ativa (escritor), e a outra, como extremidade passiva (leitor). O *endpoint* ativo é o ponto final que origina a ligação inicial e é responsável por restabelecer a ligação com o outro *endpoint*, já o passivo é o ponto final que aguarda para receber uma solicitação de conexão. Em termos de terminologia cliente/servidor, o *endpoint* ativo é o cliente e o ponto de extremidade passiva é o servidor. Para conectar dois pontos finais e criar um fluxo de rede válido, você deve especificar a URL (*Uniform Resource Locator*) de um terminal remoto então, ao criar *endpoint* ativo, deve especificar a URL do terminal remoto que irá se conectar ao *endpoint* passivo (NATIONAL INSTRUMENT, 2013). Os blocos de fluxo de redes no LabVIEW são o escritor e o leitor, conforme figura 9.

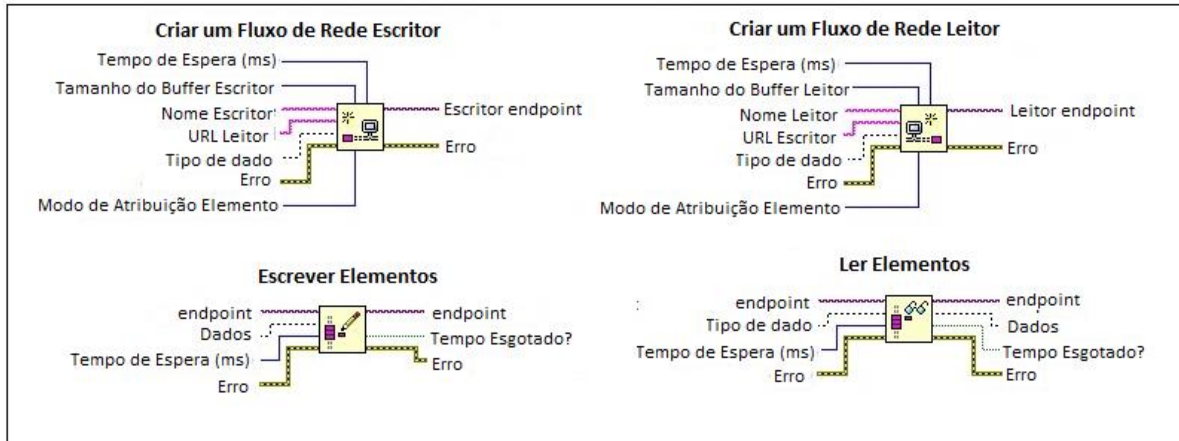


Figura 9 - Bloco de fluxo de rede em LabVIEW

Fonte: Autoria própria.

Um esquema de utilização dos blocos de fluxo de rede é descrito na figura 10, onde (1) cria o fluxo de rede escritor no computador 1 e leitor no computador 2. Em (2), o *endpoint* escritor estabelece uma conexão com o terminal leitor usando a URL do terminal leitor. Em (3) os elementos são gravados continuamente dentro do *While Loop*. O fluxo é lido continuamente em (4). O bloco (5) faz a verificação se todos os elementos foram lidos pelo terminal leitor. Após a verificação o *endpoint* do escritor é destruído pelo terminal (6). Ao destruir o *endpoint* do leitor (7) o fluxo é destruído.

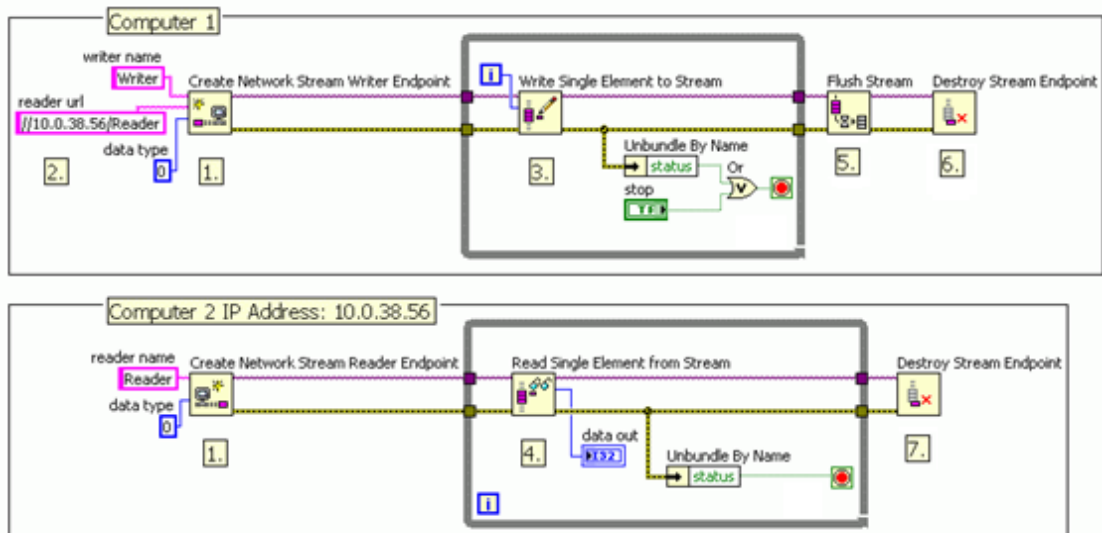


Figura 10 - Exemplo de comunicação de fluxo de rede

Fonte: Adaptado National Instrument (2010).

Fluxos de rede não são determinísticos, podendo ocasionar *jitter*, medida em que o tempo de execução não cumpre o tempo estabelecido, nos *loops* de tempo crítico. Para manter o determinismo do ciclo é aconselhável que a aquisição dos dados estejam em um *loop* de tempo crítico e o código do fluxo de rede em um *loop* de baixa prioridade. Para enviar os dados entre os dois *loops* de maneira determinística, é necessário utilizar um método de comunicação entre processos (figura11), o RT FIFO é um destes métodos (NATIONAL INSTRUMENT, 2013).

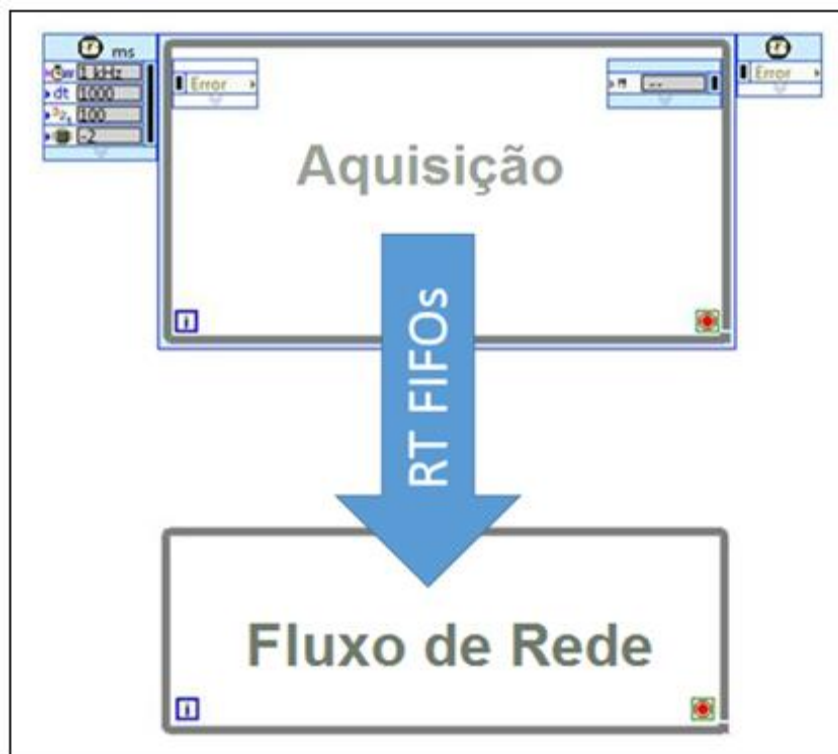


Figura 11 - Comunicação entre processos.

Fonte: Autoria própria.

2.1.3.3 RT FIFO

Para a transferência de dados entre dois circuitos paralelos de maneira determinista, é necessária a utilização de RT- FIFOs para assegurar que cada ponto de dados é lido uma vez, de forma a evitar a perda de dados.

RT FIFOs possuem este nome devido à abreviação do termo inglês *first-in-first-out*, ou seja, primeiro a entrar é o primeiro a sair. São *buffers* de memória que têm comportamento determinístico. Sua implementação pode ser feita utilizando blocos, que são semelhantes a filas onde o dado entra e quando solicitado sai, ou com uma

variável compartilhada com RT FIFO habilitado. A implementação de variáveis compartilhadas com RT FIFO habilitado é mais simples do que a programação dos blocos destas funções, porém, os blocos permitem um melhor controle. Por exemplo, com os blocos RT FIFO, você pode decidir quando os FIFOs são criados e quando eles são eliminados (NATIONAL INSTRUMENT, 2012).

Após o tempo limite especificado no terminal, o valor mais antigo do RT FIFO é substituído. Se o FIFO não tem espaço disponível e o valor do tempo de espera expira, ocasiona a substituição e perda do elemento mais antigo. Por isso, a taxa de amostragem e a dimensão da amostra são importantes na escolha do tamanho do FIFO (NATIONAL INSTRUMENT, 2012).

Os blocos do RT FIFO no LabVIEW são apresentados na figura 12, e um esquema do funcionamento do RT FIFO é mostrado na figura 13, na qual (1) cria a RT FIFO, em (2) os dados da aquisição são escritos no RT FIFO e enviados para o *loop* de baixa prioridade, onde os dados da RT FIFO são lidos pelo fluxo de rede (3) e em (4) as referências da RT FIFO são deletadas.

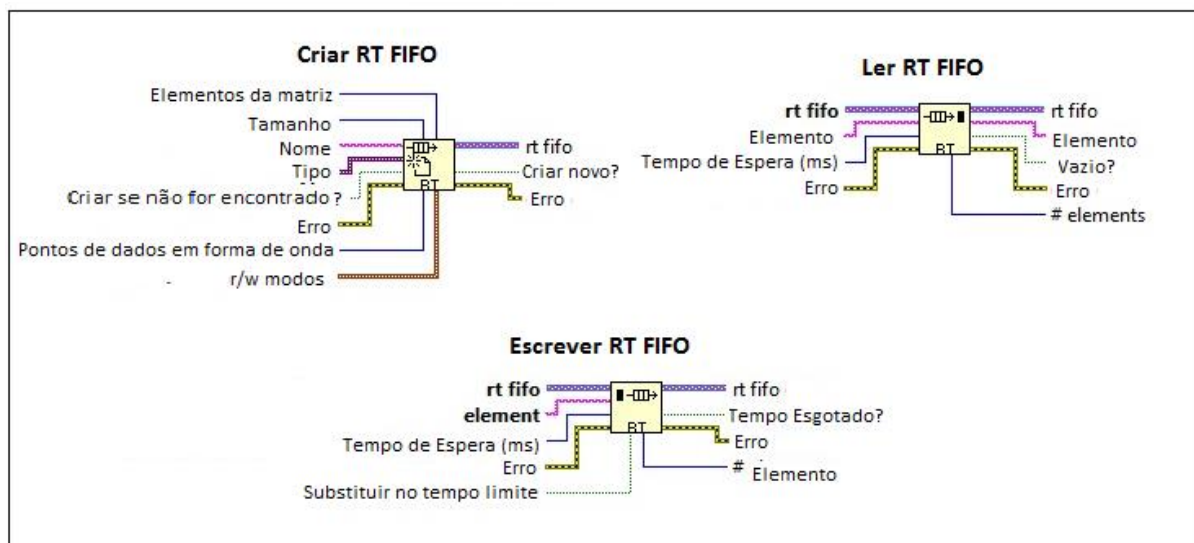


Figura 12 - Bloco RT FIFO em LabVIEW

Fonte: Autoria própria.

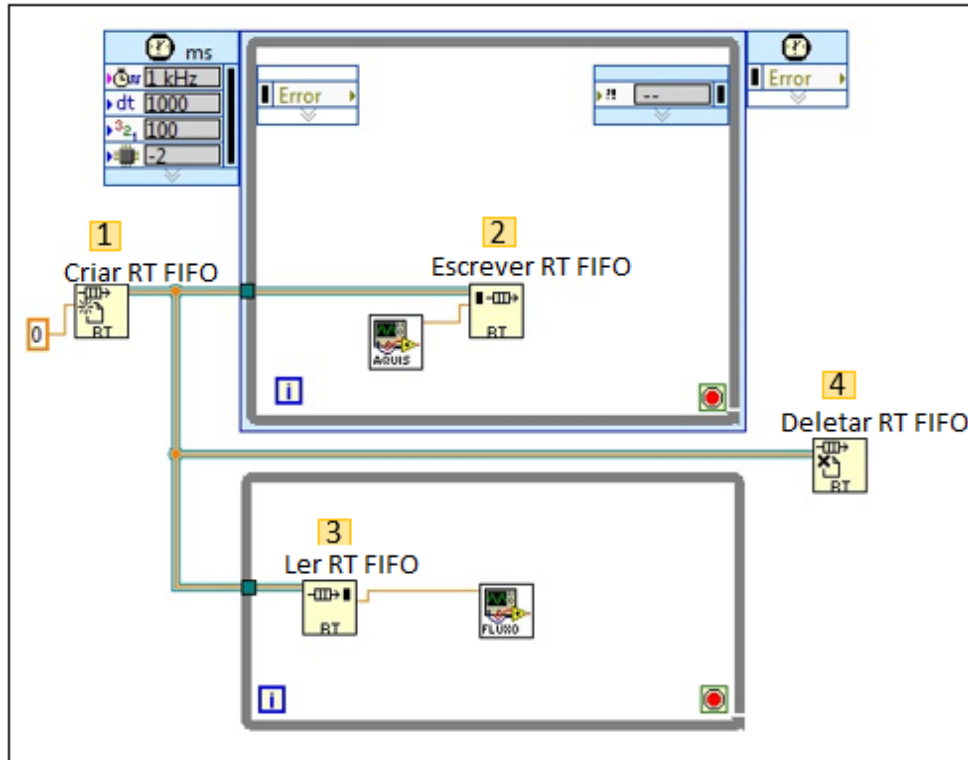


Figura 13 - Esquema de funcionamento RT FIFO

Fonte: Autoria própria.

2.2 SISTEMA DE AQUISIÇÃO

A aquisição de dados (DAQ) está presente em diversos tipos de sistema que coletam informações e auxiliam no processo de tomada de decisão. Adquirir dados é o processo de medir informações do sistema real e apresentar ao observador os valores das variáveis medidas. Geralmente o valor medido é diferente do valor real, ou seja, o valor da entrada é o valor real e sua saída é o valor medido pelo sistema (BATISTA, 2006).

A maioria dos dados medidos é de natureza analógica, variam continuamente dentre de uma faixa de valores, e necessitam de uma conversão para sinais elétricos. Para a conversão e tratamento dos sinais são utilizados sensores e condicionadores, fornecendo sinais apropriados à placa de aquisição de dados. Após condicionados na forma desejada, os sinais podem ser lidos pelo computador com placas específicas instaladas (BATISTA, 2006).

Ao construir um sistema de aquisição que utiliza um computador desktop, os resultados dependerão do computador, dos sensores, do condicionamento de sinais e dos equipamentos e programas utilizados.

A programação e aquisição dos dados podem ser feitas com o auxílio de programas específicos, como o LabVIEW e placas de aquisição da National Instruments, que trabalham com o conceito de instrumento virtual (VI) e que tem como base o uso de um computador para desenvolver um aplicativo, simulando instrumentos físicos com a versatilidade de um programa de aquisição (GARCIA et al., 2003).



Figura 14 - Partes de um sistema de aquisição

Fonte: National Instrument (2015).

A figura 14 mostra um modelo de sistema de aquisição utilizando um computador. Os sensores medem os fenômenos físicos e os transformam em um sinal elétrico mensurável. O dispositivo DAQ, o *hardware* (placa de aquisição), é a interface entre o sistema e o computador, ele possui circuitos eletrônicos de condicionamento de sinais que condiciona o sinal em um formato adequado para ser enviado ao conversor analógico digital. Após o sinal ser convertido para o formato digital, é enviado através do barramento para o computador. O computador deve ter *softwares* específicos para comunicação com a placa e também para aquisição, análise e apresentação dos dados da medição ao usuário.

2.3 SISTEMA DE CONTROLE

Controlar um processo é um papel fundamental nos processos industriais e de produção, produzindo meios de otimizar o desempenho de sistemas dinâmicos. Os sistemas de controle tratam da manutenção de grandezas em valores desejados ou condução de uma variável a um determinado valor (PINHEIRO, 2009).

A técnica de controle consiste em obter os dados da saída do sistema, por meio de sensores. Com estes dados, o controlador toma as decisões necessárias, comparando as saídas com um valor pré-definido, e envia os comandos necessários ao atuador, o qual ajusta uma ação para correção do erro com o intuito de melhorar o desempenho do sistema e obter uma variável de saída o mais próximo possível do desejado (BARCESSAT; TABARANA, 2006). A figura 15 demonstra uma representação deste sistema onde V_p é a variável controlada e SP é o *setpoint*.

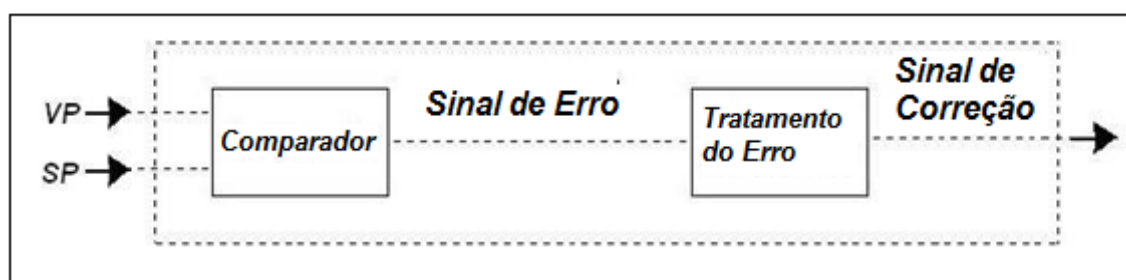


Figura 15 - Controlador em blocos

Fonte: Silva (2008).

Com isso vários parâmetros estão associados ao sistema de controle. Os quais fazem alusão às variáveis existentes em um sistema e relacionam diretamente ao funcionamento desejado deste. Essas grandezas são denominadas variáveis controladas e variáveis manipuladas. A variável controlada, ou variável de saída, é qualquer variável que se queira regular e a atuação nessas variáveis permite que a variável controlada seja mantida próxima ou igual a um ponto de operação (*setpoint*). A variável manipulada, ou variável de entrada, é a referência para a variável de saída, na qual se deseja manter um valor determinado (*setpoint*). Qualquer evento que retire o sistema do seu ponto de operação desejado é denominado distúrbio (BARCESSAT; TABARANA, 2006).

Os sistemas de controle podem ser divididos em duas estruturas, dependendo da complexidade e das possíveis perturbações que o sistema pode sofrer. Para sistemas simples com poucas perturbações, o controle em malha aberta é indicado, porém, quando o sistema apresenta distúrbios e necessita de um controle próximo do exato, o controle de malha fechada é indicado. O Sistema de Controle em Malha Aberta é um sistema onde os valores de entrada não são afetados pelos valores de saída. Caso ocorra algum erro, nenhum ajuste será feito por falta de informação do controlador

sobre a saída do sistema. Os Sistemas de Controle em Malha Fechada são sistemas realimentados, onde a variável de saída é comparada com um valor de referência (*setpoint*), as diferenças entre os dois valores resulta em um erro. A verificação do erro atuante e a correção deste ocorrem através de uma determinada ação de controle (ROCHA, 2009).

A figura 16 (a) ilustra um sistema de controle em malha aberta e (b) um sistema em malha fechada.

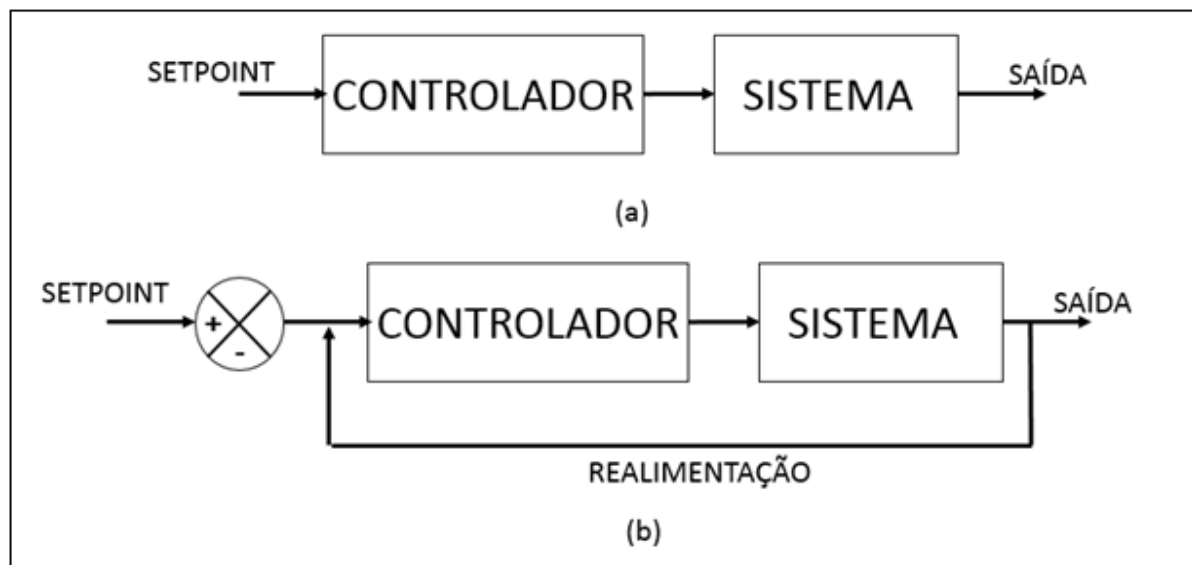


Figura 16 - (a) Sistema em malha aberta; (b) Sistema em malha fechada

Fonte: Autoria própria.

Um sistema em malha fechada, altamente empregado na indústria, são os controladores PID. Estes controles são utilizados quando o modelo matemático da planta não é conhecido, portanto, o emprego de alguns métodos de projeto analítico não pode ser utilizado (PINHEIRO, 2009).

A técnica de controle PID consiste em combinar ações proporcionais, integrais e derivativas, gerando um sinal de controle, a fim de garantir um controle estável e preciso.

2.3.1 Controle Proporcional

A ação de controle proporcional é uma das formas mais simples de controle, onde o sinal de controle aplicado apresenta uma relação matemática proporcional à

amplitude do valor do erro. Sendo assim, este controlador é um amplificador, no qual a amplitude de correção é proporcional a amplitude do desvio (SILVA, 2008). Ele pode ser representado pela equação (1):

$$u(t) = K_p e(t) \quad (1)$$

Onde K_p é o ganho proporcional e $e(t)$ é o erro entre o sinal de entrada (*setpoint*) e o sinal de saída $y(t)$:

$$e(t) = r(t) - y(t) \quad (2)$$

Aplicando Laplace e transformando para o domínio complexo à equação (1), obtém-se:

$$U(s) = K_p E(s) \quad (3)$$

Sua representação em diagrama de bloco pode ser vista na figura 17.

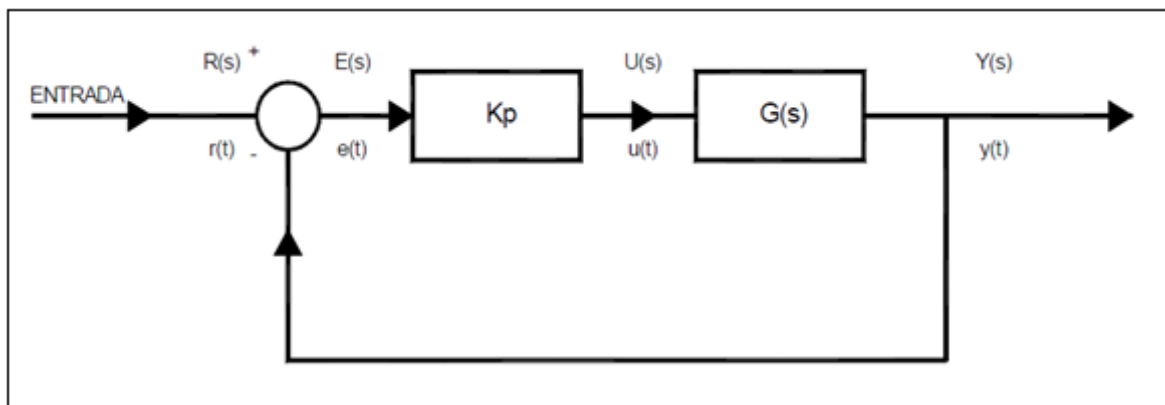


Figura 17 - Diagrama de blocos controle proporcional

Fonte: Barcessat e Tabarana (2006).

No controle proporcional, o sinal de saída é K vezes maior que o sinal de entrada. Porém, este sinal pode assumir qualquer valor, contanto que esteja dentro dos limites de saída máxima e mínima. Quando o valor de saída atinge esses limites o sistema está saturado. Valores acima do limite superior estão com 100% da capacidade do atuador, e valores abaixo do limite inferior estão com 0% de sua capacidade. A região entre estes limites é chamada de banda proporcional do sistema.

O tamanho da banda proporcional influencia a ação proporcional de controle: quanto menor o seu valor maior sua influência na correção do erro (TRAJANO CAMARGO, 2012).

2.3.2 Controle Proporcional e Integral

Os controladores com ação integral são controles de ação dinâmica, pois sua ação varia em função do tempo enquanto existir o erro (PINHEIRO, 2009).

Sua saída é proporcional à integração do erro ao longo do tempo:

$$u(t) = \frac{1}{T_i} \int_0^t e(t) dt \quad (4)$$

Aplicando Laplace e transformando para o domínio complexo à equação (4), obtém-se:

$$U(s) = \frac{E(s)}{sT_i} \quad (5)$$

Quando utilizado isoladamente, apresenta uma piora das características dinâmicas do sistema. Porém, em conjunto com o controle proporcional, melhora o comportamento do sistema.

A equação (6) apresenta a ação de controle proporcional e integral:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (6)$$

A equação (6) no domínio complexo:

$$U(s) = K_p E(s) + \frac{K_p}{sT_i} E(s) \quad (7)$$

Quando o erro é grande, a ação proporcional atua no erro predominando sobre o integrador, à medida que o sistema aproxima do valor desejado e o erro diminui a ação proporcional suaviza sua atuação, predominando a ação do integrador. A adoção

de uma ação integral excessivamente atuante aumenta as oscilações e leva o sistema a instabilidade (TRAJANO CAMARGO, 2012).

A figura 18 ilustra o diagrama de blocos de um controlador Proporcional – Integrador

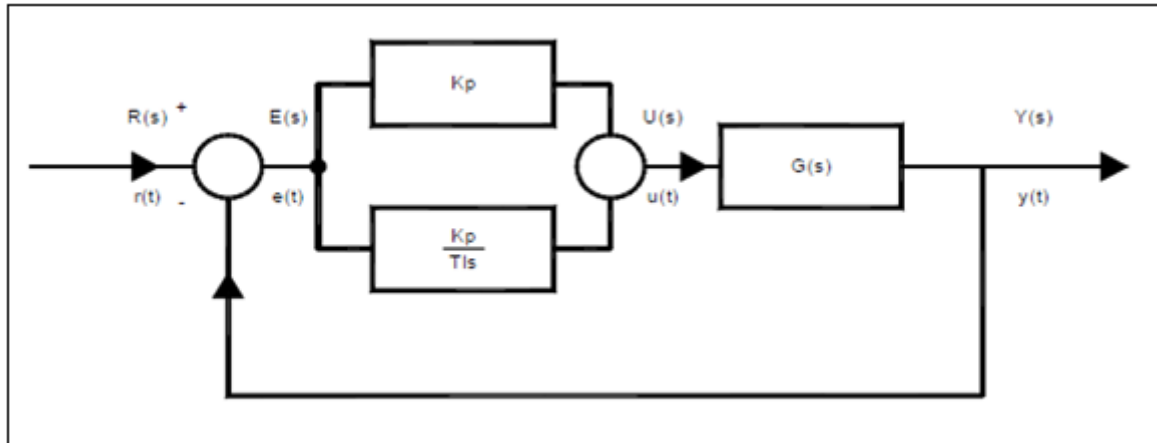


Figura 18 - Diagrama de blocos do controle proporcional e integral

Fonte: Barcessat e Tabarana (2006).

2.3.3 Controle Proporcional Integral e Derivativo

Nos controladores de ação derivativa, a saída do controlador é proporcional à derivada do sinal do erro, ou seja, proporcional à variação do erro (TRAJANO CAMARGO, 2012).

Caso o erro varie muito rápido, ele irá atuar para eliminar esta variação. A equação (8) apresenta a ação de controle derivativa:

$$u(t) = T_d \frac{d e(t)}{dt} \quad (8)$$

A ação derivativa é usada para corrigir a resposta transitória do sistema, sendo sensível a ruídos os quais podem ser confundidos com uma transição do sistema. O derivativo nunca é utilizado sozinho, pois ele apenas atua quando o erro varia rapidamente. Se o erro está variando lentamente, o sinal de saída deste controlador será baixo (TRAJANO CAMARGO, 2012).

Utilizando em conjunto com o proporcional e o integrador, obtém-se a equação (9):

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{d e(t)}{dt} \quad (9)$$

A equação (9) no domínio complexo:

$$U(s) = K_p E(s) + \frac{K_p}{sT_i} E(s) + sK_p T_d E(s) \quad (10)$$

A união da ação proporcional, integral e derivativa é o chamado Controlador PID, na figura 19 o diagrama de blocos é apresentado. Com um controlador PID, podemos obter outros controladores (P, PI ou PD), zerando-se aquele que não interessa. Exemplo: se queremos um controlador PI, basta zerar o ganho do Derivativo e assim por diante (TRAJANO CAMARGO, 2012).

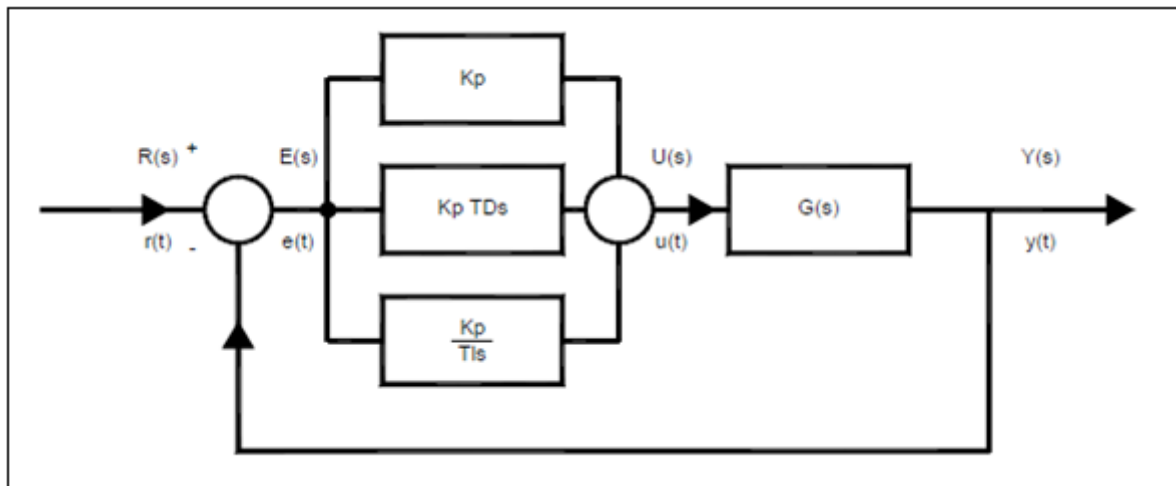


Figura 19 - Diagrama de blocos do controle proporcional, integral e derivativo

Fonte: Barcessat e Tabarana (2006).

2.3.3.1 PID em LabVIEW

O *software* LabVIEW possui em sua biblioteca, um bloco baseado no método de controle PID. A figura 20 apresenta o bloco utilizando PID. Este bloco facilita a programação, pois diminui o tempo, já que as funções estão todas implementadas. O programador apenas seleciona a referência (*setpoint*) e a variável manipulada, e envia a saída do controle para o sistema a ser controlado.

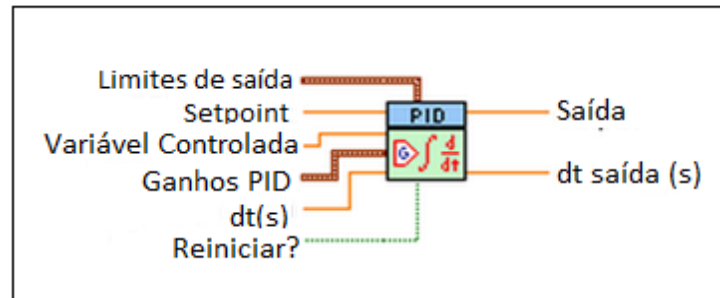


Figura 20 - Bloco PID em LabVIEW

Fonte: Autoria própria.

O bloco PID cria um controle no painel frontal, conforme figura 21, onde o usuário pode sintonizar os ganhos.

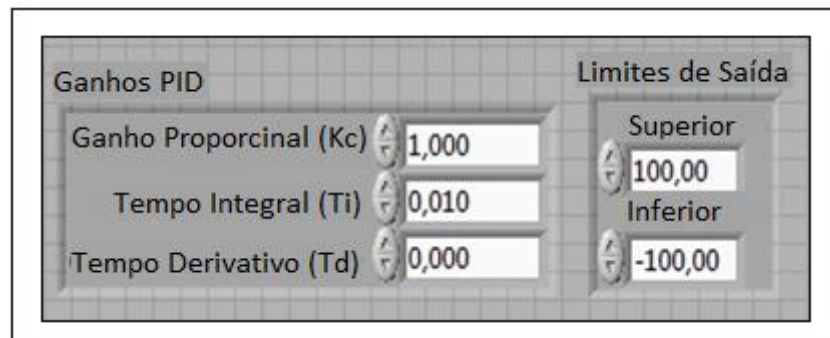


Figura 21 - Painel frontal do bloco PID em LabVIEW

Fonte: Autoria própria.

2.4 SERVOMOTOR

O Servomotor é um dispositivo eletromecânico, como muitas outras máquinas síncronas, que tem o posicionamento angular do seu eixo a partir de um sinal elétrico. Possuem grande aplicabilidade no setor da robótica onde o controle da posição do rotor é de extrema importância. O controle da posição é feito através de sensores acoplado ao eixo do rotor, como *encoder*. Os *encoders* são dispositivos largamente utilizados na indústria, capaz de medir com precisão o deslocamento angular.

2.4.1 Encoder

Os *encoder* são dispositivos que convertem o movimento rotacional do eixo, movimentos angulares, em pulsos elétricos. Sendo considerados como transdutores de movimento (MATOS, 2012).

Esses pulsos elétricos são gerados para um determinado incremento de rotação e posteriormente transformados em sinais digitais que podem ser trabalhados e convertidos em grandezas conhecidas, como a velocidade. A velocidade não é medida diretamente pelo *encoder* e sim a posição, porém tendo a distância percorrida através da contagem dos pulsos e o tempo necessário para obter essa distância, a velocidade pode ser calculada (ANDRADE, 2008).

Há vários tipos de *encoder* como o magnético, contato, resistivo e óptico. Sendo o mais utilizado o *encoder* óptico (figura 22). Este sensor possui um disco com ranhuras acoplado ao eixo, que se move entre uma fonte de luz, que pode ser um diodo emissor de luz (LED), e um detector. Quando o eixo começa a girar a luz atravessa as áreas transparente e é interrompida quando encontra as ranhuras, o detector captura esses sinais gerando uma onda pulsante (ANDRADE, 2008).

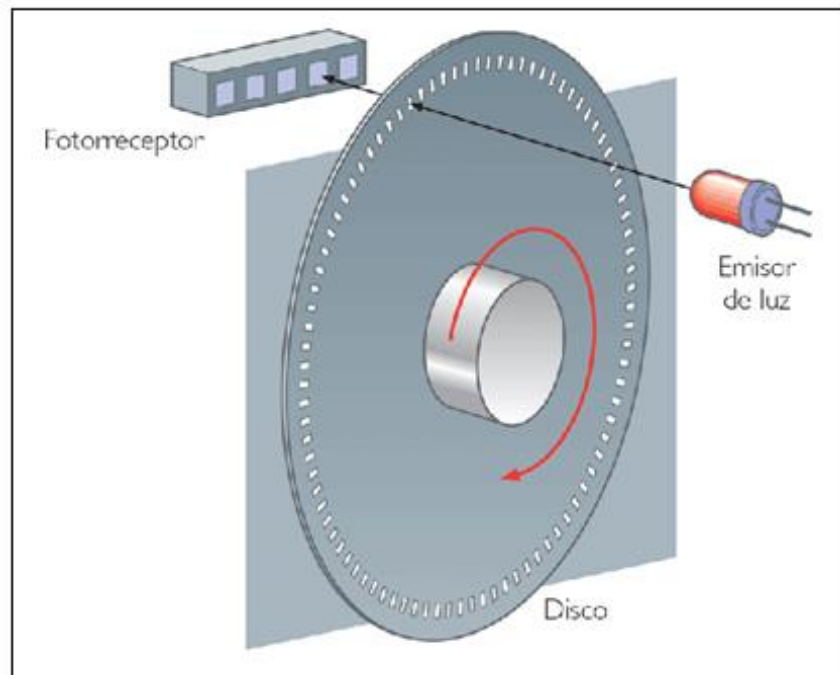


Figura 22 - Encoder Óptico

Fonte: Teleco (2015).

Os *encoders* ópticos podem ser divididos em incremental ou absoluto. O *encoder* incremental, figura 23, também conhecido como em quadratura, possui dois canais A e B igualmente espaçados, que permitem a detecção da posição, e defasados permitindo avaliar o sentido de rotação. O ponto z é a referência onde indica o final de uma volta e o início da contagem (MATOS, 2012).

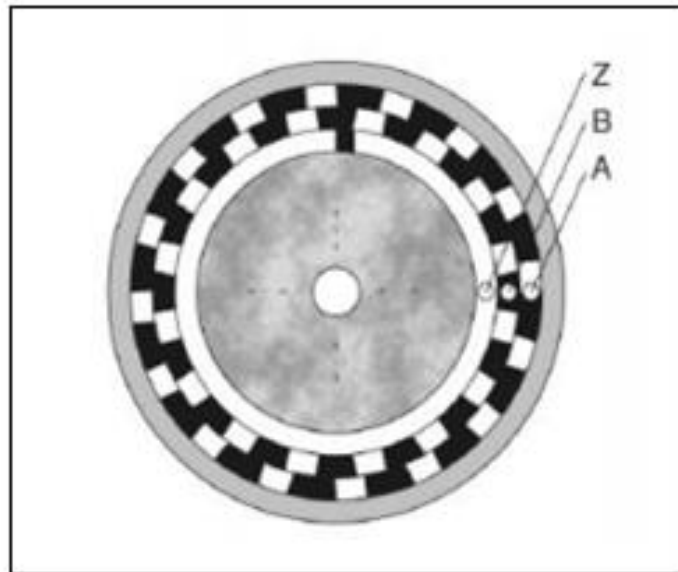


Figura 23 - Encoder Incremental

Fonte: Matos (2012).

O *encoder* absoluto possui apenas um canal e fornecem um valor numérico específico para cada posição angular. Nos *encoders* absolutos, cada posição corresponde a uma combinação de sinais (A1, A2, A3, A4, A5, A6), conforme figura 24 (MATOS, 2012).

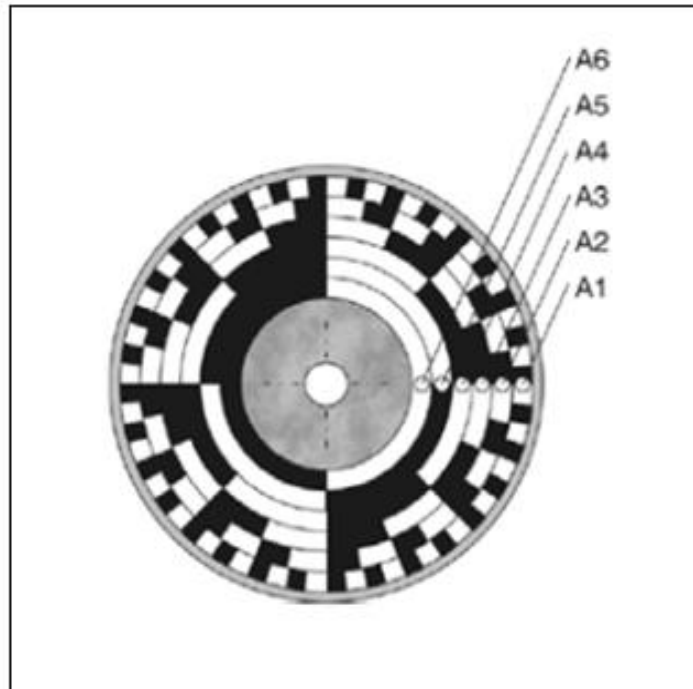


Figura 24 - Encoder Absoluto
Fonte: Matos (2012).

Para realizar medições com *encoders*, são utilizados contadores. O contador fornece um valor que representa o número de bordas contadas, transições de alto para baixo, com base nos sinais de entrada. O contador incrementa ou decrementa a contagem conforme o estado (alto ou baixo) (NATIONAL INSTRUMENT, 2013). Há três tipos básicos de codificação: X1, X2 e X4.

A codificação X1 ocorre na borda de subida do canal A se A estiver na frente de B, e na borda de descida do canal A se B estiver a frente (NATIONAL INSTRUMENT, 2013). Conforme figura 25.

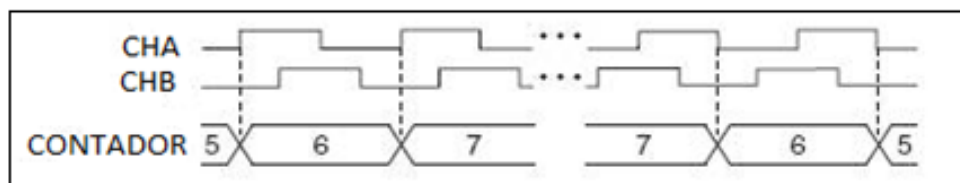


Figura 25 - Codificação X1
Fonte: National Instrumet (2013).

A codificação X2 ocorre da mesma maneira, porem o contador é incrementado ou decrementado em todas as bordas do canal A. Cada ciclo resulta em dois incrementos ou decrementos, figura 26 (NATIONAL INSTRUMENT, 2013).

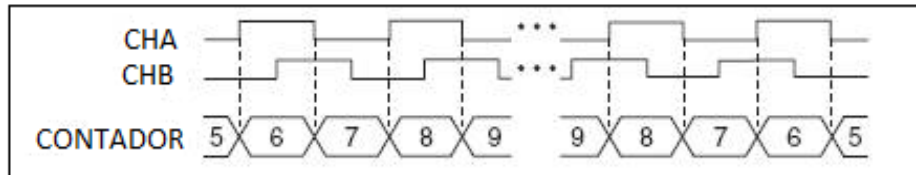


Figura 26- Codificação X2

Fonte: National Instrumet (2013).

O contador é incrementado ou decrementado de maneira similar em cada borda dos canais A e B na codificação X4, ou seja, o contador é incrementado ou decrementado na borda de subida e descida de ambos canais. Cada ciclo resulta em quatro incrementos ou decrementos, como mostrado na figura 27 (NATIONAL INSTRUMENT, 2013).

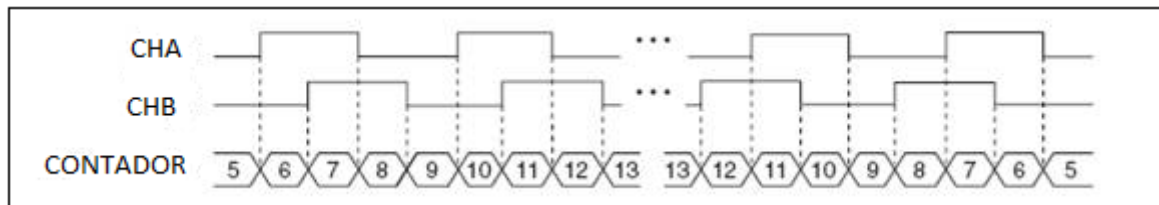


Figura 27 - Codificação X4

Fonte: National Instrumet (2013).

2.4.2 Modulação de Largura de Pulso

Modulação de largura de pulso ou PWM (*Pulse Width Modulation*) tem como base o mesmo princípio de um interruptor de ação rápida e uma tensão na carga que deve ser controlada, conforme figura 28.

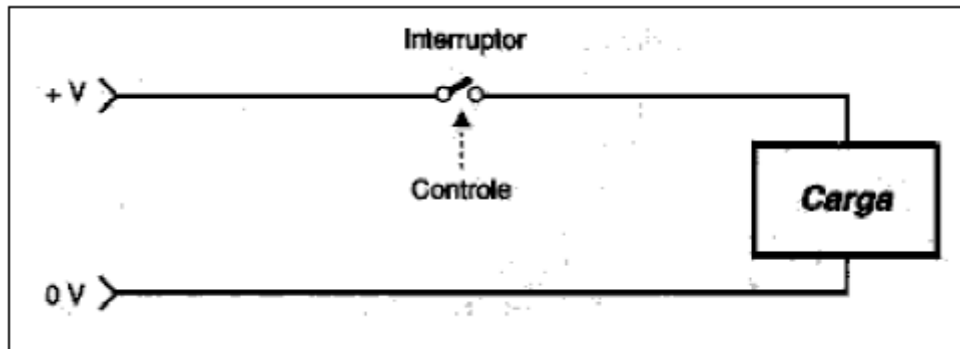


Figura 28 - Circuito de controle de corrente na carga
Fonte: Silva (2008).

Ao abrir o interruptor a potência aplicada na carga é nula, quando o interruptor é fechado a potência aplicada é máxima. Para obter uma potência intermediária, a chave precisa ser aberta e fechada rapidamente, ficando 50% do tempo aberta e 50% fechada, resultando em uma tensão de 50%. O tempo que o interruptor permanece fechado define a largura de pulso e o intervalo entre pulsos é o tempo em que ele permanece aberto. A soma desses dois tempos é o período do sistema. Ao variar a largura do pulso e o intervalo entre eles, de zero até o máximo obtém-se uma potência que varia na mesma proporção (NASSER et al., 2014). A figura 29 apresenta a tensão na carga variando no tempo, conforme a largura do pulso.

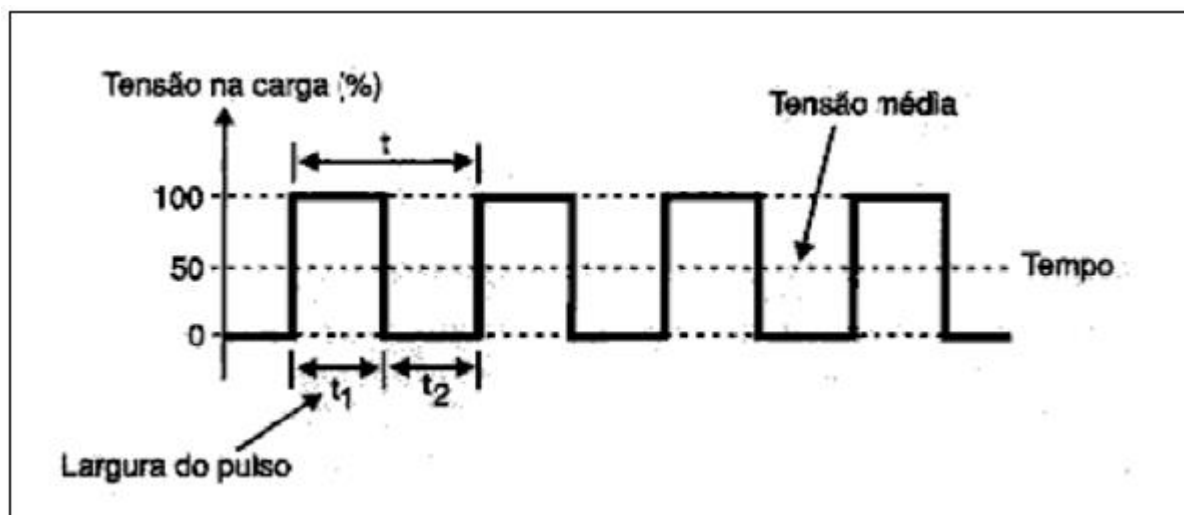


Figura 29 - Variação da largura de pulso
Fonte: Silva (2008).

Um sinal de PWM é composto por dois componentes principais que definem o seu comportamento: a largura de pulso e a frequência (MOREIRA et al, 2010). Utilizando estes dois parâmetros na equação (11) pode-se obter o *duty cycle*:

$$DC = \frac{LP}{P} \times 100\% \quad (11)$$

O *duty cycle* é a porcentual de tempo em que o sistema está ligado em comparação com o tempo em que está desligado.

O módulo de largura pulso, PWM, são usados em várias aplicações de controle. A sua principal utilização é para o controle da tensão média nos terminais do motor, e com isso controlar a corrente e o torque do motor CC.

3 METODOLOGIA E MATERIAIS

Neste capítulo será discutido os materiais utilizados na confecção do kit além dos métodos para a elaboração do programa e implementação do sistema.

3.1 COMPONENTES DO KIT

O kit possui os seguintes componentes:

- Servomotor
- *Encoder* incremental em quadratura
- Placa de decodificação de quadratura
- Gerador acoplado através de uma correia ao motor, para reproduzir cargas mecânicas.
- Driver de potência.
- Placa *pull up*.

A figura 30 apresenta todos os componentes do kit.

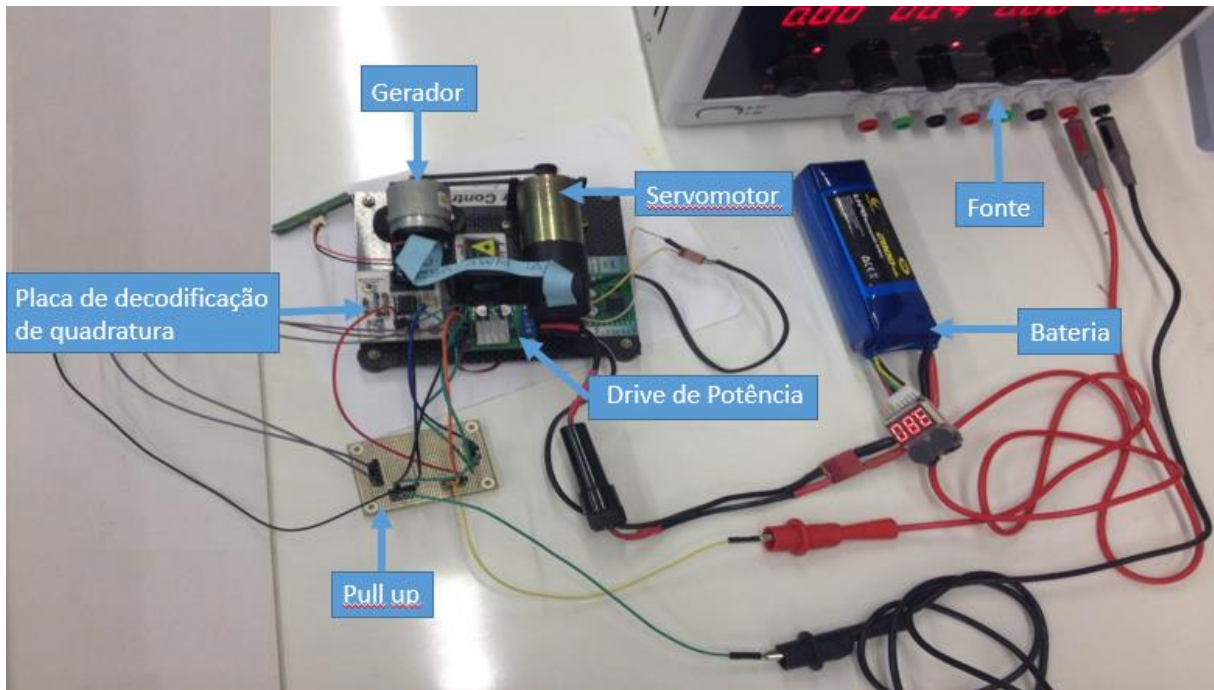


Figura 30 - Sistema completo

Fonte: Autoria própria.

3.1.1 Servomotor

O servomotor para controle foi um motor CC da fabricante Pittman, com tensão de trabalho máxima de 19,1V e rotação máxima de 10.000 rpm.

Para realizar a leitura da posição angular do eixo do motor, foi utilizado um *encoder* incremental em quadratura para leitura de posição e determinação da velocidade, modelo HEDS-5310 da marca Avago, que possui 512 linhas e 3 canais.

Quando se deseja simular cargas mecânicas em motores, é necessário acoplar um gerador, através de uma correia, ao eixo do motor. Como pode ser visto na figura 31. Dessa forma é possível controlar a carga mecânica no eixo do motor conforme a variação da corrente no gerador, para que ocorra esta variação da corrente foi utilizada uma resistência de 47R e 10W no gerador.

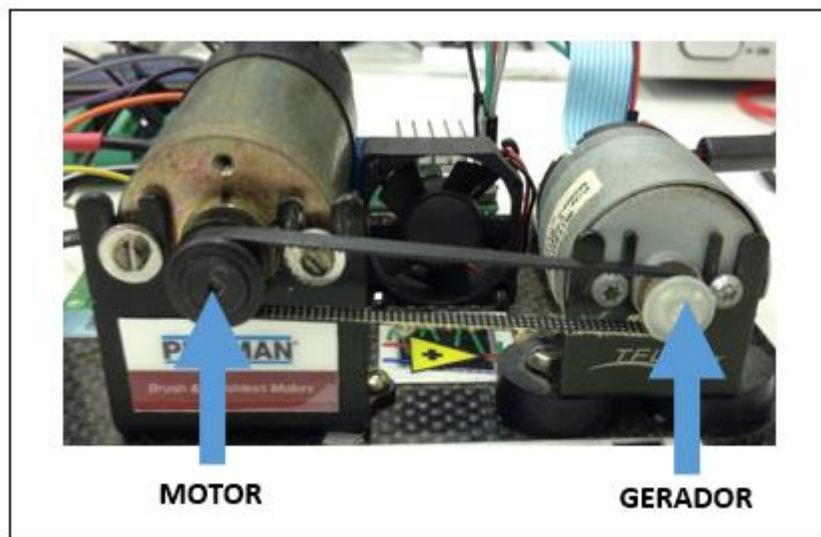


Figura 31 - Gerador acoplado ao servomotor

Fonte: Autoria própria.

3.1.2 Driver de potencia

O acionamento do servomotor ocorre através do drive de potência da fabricante Pololu. O driver possui um microcontrolador, MC33887, produzido pela Freescale Semiconductor, de fácil integração com um motor CC com tensão de 5-28 V e corrente até 5 A.

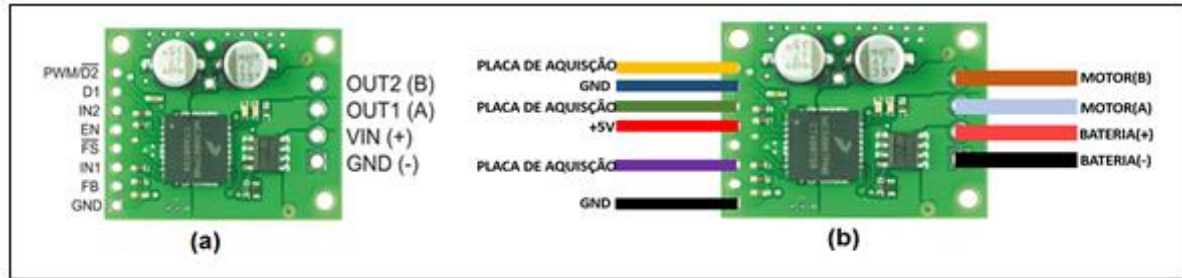


Figura 32 - (a) Pinos do driver de potência (b) Ligação dos pinos

Fonte: Autoria própria.

A descrição dos pinos pode ser vista na figura 32 (a), tendo as entradas do lado esquerdo e as saídas do lado direito do driver de potência. A ligação de cada pino é mostrada na figura 32 (b), onde o PWM se liga a placa de aquisição para controlar a velocidade do motor, D1 está em GND, IN1 e IN2 também estão conectados com a placa para controle da direção do motor, o pino EN é alimentado com +5 V por uma fonte e os pino de realimentação (FB) e de segurança (FS) não foram utilizados. Os canais dos motores A e B estão ligados na Saída 1 e 2 respectivamente e a alimentação do motor ocorre através de uma bateria de 15,5V.

3.1.3 Placa de decodificação em quadratura

A placa foi desenvolvida no Laboratório de Sistemas Automatizados e Controle, ela recebe os dados do *encoder* localizado no servomotor e transforma esses dados para serem enviados para a placa de aquisição. Os dados do *encoder* são as entradas e os canais, A e B, as saída conforme a figura 33. A placa é alimentada por uma fonte de 5V.

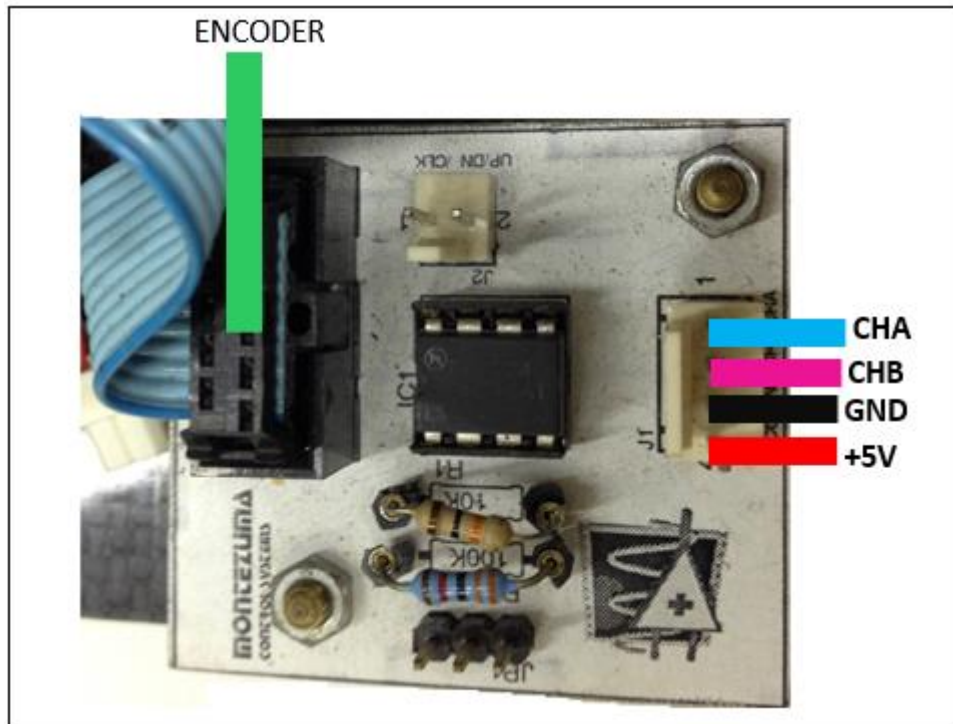


Figura 33 - Placa de decodificação de quadratura

Fonte: Autoria própria.

Ao realizar os experimentos, percebeu que ocorria flutuações nos canais de saída, com isso foi confeccionada uma placa de *pull up*, com resistores de 3300 ohms para evitar flutuações nos valores de saída, garantindo o estado da porta configurada como entrada digital em nível lógico 0 ou 1. Essa placa foi elaborada no *software* Eagle (figura 34) e confeccionada no laboratório.

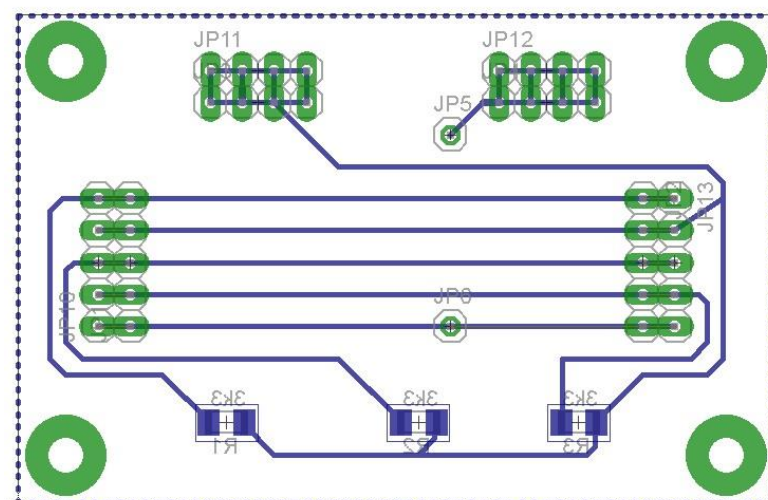


Figura 34 - Placa pull up elaborada no Eagle

Fonte: Autoria própria.

Os sinais da placa de decodificação de quadratura são ligados a placa *pull up*, na figura 35 são apresentadas estas ligações. Os canais A e B da placa de decodificação passa pelo resistor, contido na placa de *pull up*, e depois pode ser enviado para a placa de aquisição. A alimentação da placa é feita pela fonte de 5 V.

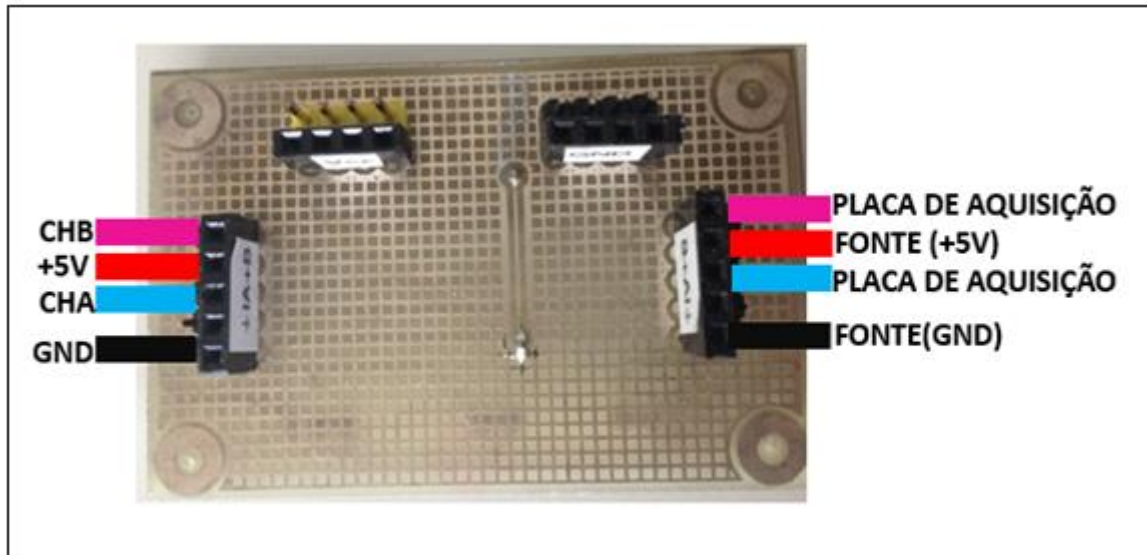


Figura 35 - Ligação Placa pull up

Fonte: Autoria própria.

3.2 SISTEMA DE AQUISIÇÃO E CONTROLE

A aquisição e o controle são realizados através da placa PCI-6221, da National Instrument, acoplada a um computador desktop (*Target*), onde o sistema fará a aquisição em tempo real, garantindo uma taxa de aquisição de 500 Hz e um tempo de resposta de 2 ms. A interação com o usuário e o armazenamento dos dados são feitas através de outro computador desktop (*Host*). O *software* de desenvolvimento da aquisição, controle e comunicação de dado utilizado foi o LabVIEW.

3.2.1 Host

O computador *host* executa o *software* de Interface Homem-Máquina que exibe e registra os dados da aquisição e também configura a ação de controle do sistema. Esta interface para o usuário (figura 36) fornece opções configuráveis na qual é

possível modificar o sinal de referência de controle (*setpoint*), ajustar os parâmetros do controle proporcional-integral-derivativo (PID), configurar a taxa de descritização do sistema e habilitar o salvamento dos dados. O sinal de referência do sistema, ou seja, a velocidade do servomotor pode ser alterada através de ondas quadradas, triangulares ou senoidais. Após configuração dos parâmetros de entrada do sistema o usuário pode verificar o desempenho do sistema de controle por meio dos gráficos da velocidade e da ação de controle.

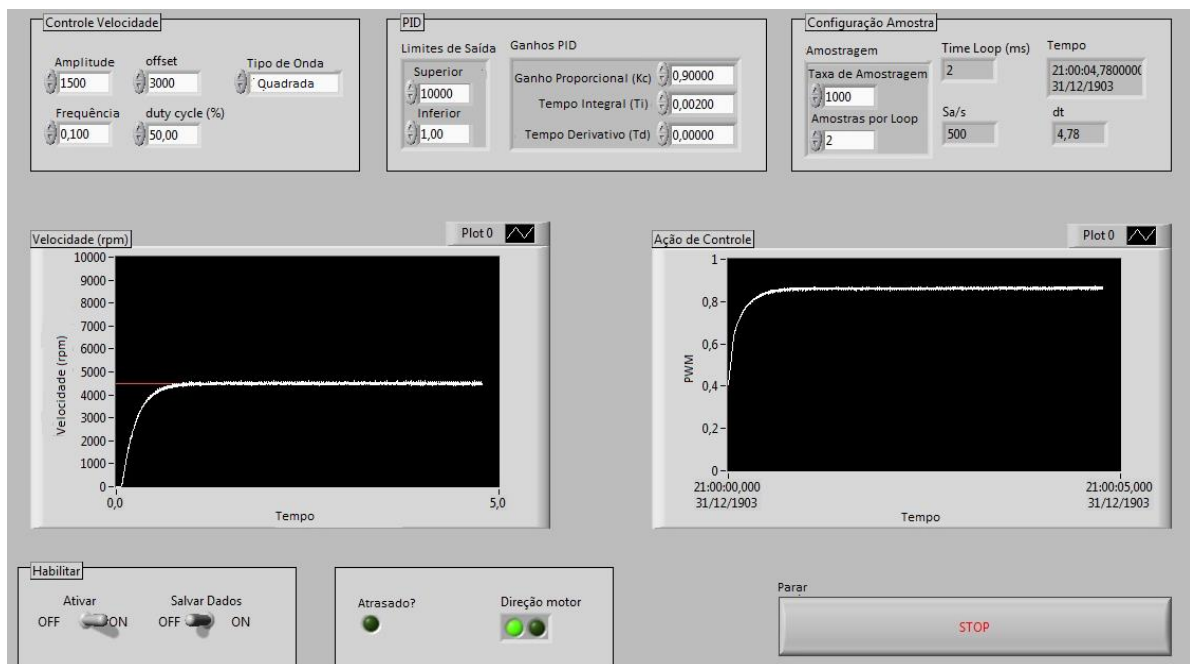


Figura 36 - Interface para o usuário

Fonte: Autoria própria.

O diagrama de blocos do programa executado pelo *host* (apêndice A) consiste no envio dos parâmetros de entradas definidos pelo usuário para o *target* e o recebimento dos dados da aquisição.

Na figura 37, os dados de entrada sendo enviados para o *target* através de Variáveis Compartilhadas Publicadas na Rede.

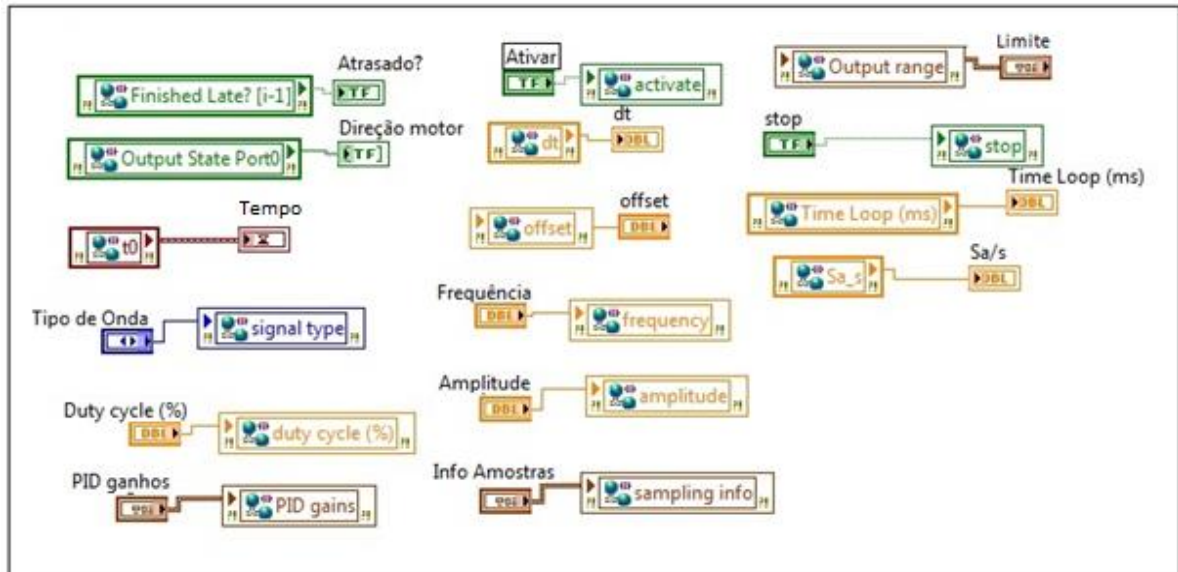


Figura 37 - Variáveis compartilhadas publicada na rede

Fonte: Autoria própria.

Os dados enviados pelo *target* são recebidos através do fluxo de rede, transformados em ondas e exibido em gráficos (figura 38). Quando habilitado o botão de salvamento dos dados, estes são armazenados no *host* conforme a figura 39.

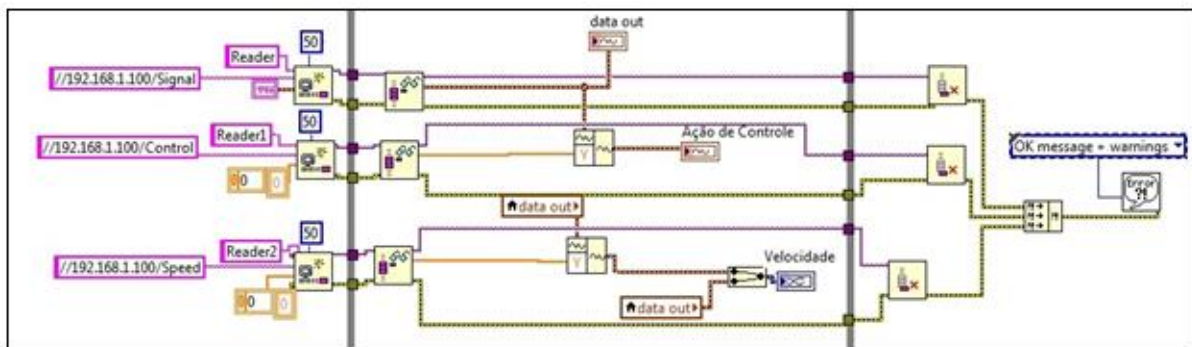


Figura 38 - Recebimento de dados através do fluxo de rede

Fonte: Autoria própria.

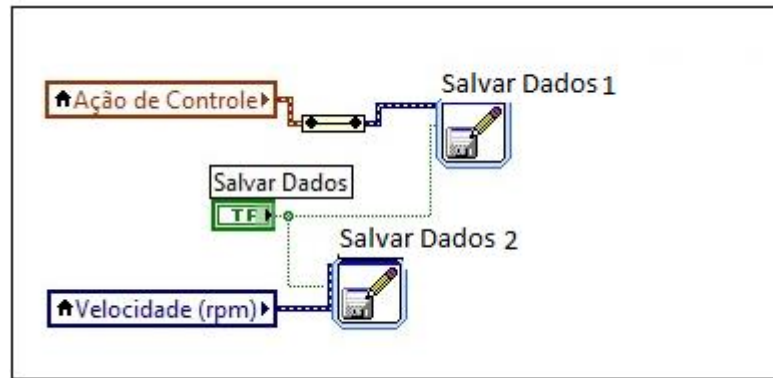


Figura 39 - Armazenamento dos dados da aquisição

Fonte: Autoria própria.

3.2.2 Target

Para realizar a aquisição em tempo real é necessária a instalação da placa de aquisição e a configuração do sistema de tempo real. A placa de aquisição é instalada no *target* conforme a figura 40, onde a placa de aquisição (1) é inserida no slot de sistema PCI (2) do computador (3).

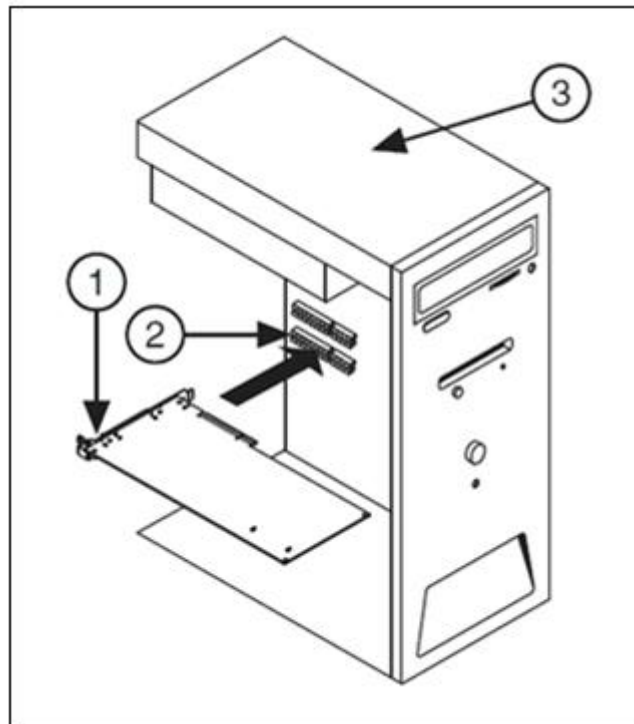


Figura 40 - Instalação da placa de aquisição

Fonte: National Instrument (2015).

Com a placa instalada é necessário configurar o sistema de tempo real, a National Instrument disponibiliza um arquivo de boot e um tutorial para formatar o computador com o sistema. Estas configurações não serão descritas neste trabalho.

No diagrama de blocos do *target* (apêndice B) os dados do *encoder*, são adquiridos, processados e enviados ao *host*. Com os parâmetros de entrada enviados pelo *host*, são feitos os devidos cálculos e enviado para o controle do sistema físico. O *target* não possui nenhum dispositivo no painel frontal, por isso não há interface para o usuário.

A configuração dos canais de entrada do sistema é feita no início do programa, conforme figura 41. O *encoder* é configurado como contador 1, com 512 linhas e codificação X4. A direção do motor é feita pelo porte digital 0. O PWM é configurado inicialmente com uma frequência de 20000Hz e um *duty cycle* de 10%.

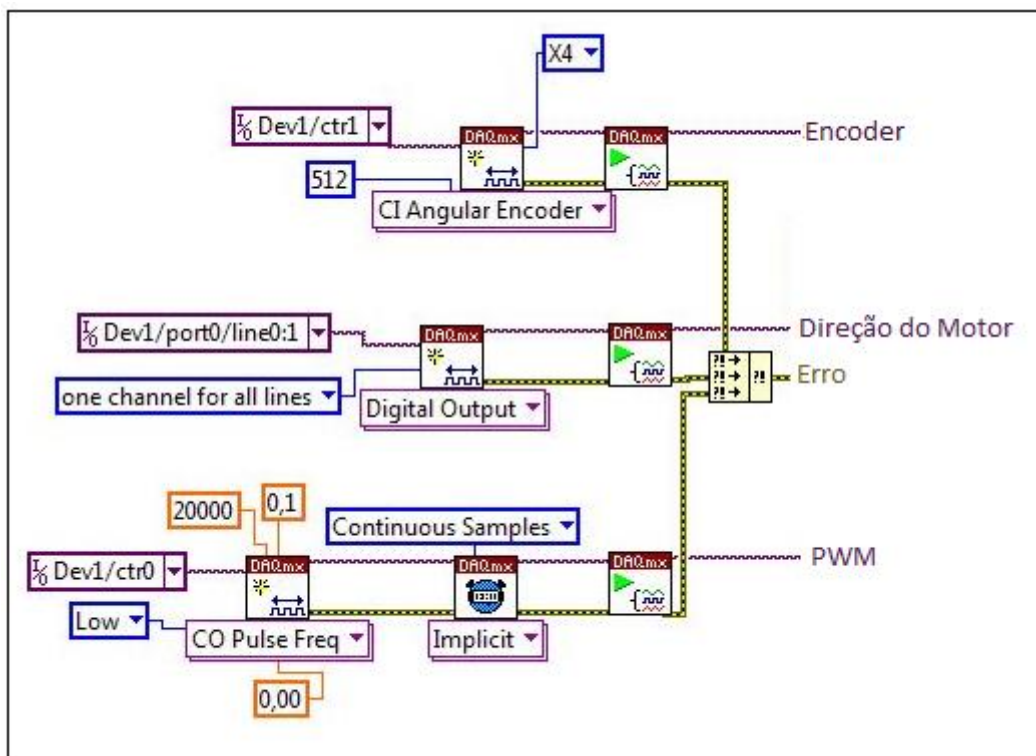


Figura 41 - Configuração dos canais de aquisição

Fonte: Autoria própria.

Para garantir que o sistema consiga enviar os dados para o *host* em tempo real, dois *loops* são executados paralelamente, um de maior prioridade, executando as tarefas de tempo crítico e outro com as tarefas não determinísticas.

A aquisição de dados no *loop* de tempo crítico, só ocorre se o botão habilitar estiver acionado, caso não esteja, o programa não executa nenhuma tarefa.

Quando habilitada a aquisição, o contador da placa aquisição os dados da posição angular, fornecida pelo *encoder*, a uma taxa de amostragem definida pelo usuário. Os dados referentes a forma de onda, definida pelo usuário, são enviados para o gerador de função, o qual forma uma onda com o valor da velocidade de referência (*setpoint*). A velocidade de referência é salva no RT FIFO e enviado para o *loop* de baixa prioridade, o seu sinal também é convertido e enviada para o subVI Somatória (apêndice C) juntamente com os dados obtidos do *encoder*, no subVI ocorre a somatória das velocidade de referência e a somatória dos dados do *encoder*, esta interações ocorrem à taxa de amostragem definida pelo usuário. Esta parte do programa pode ser vista na figura 42.

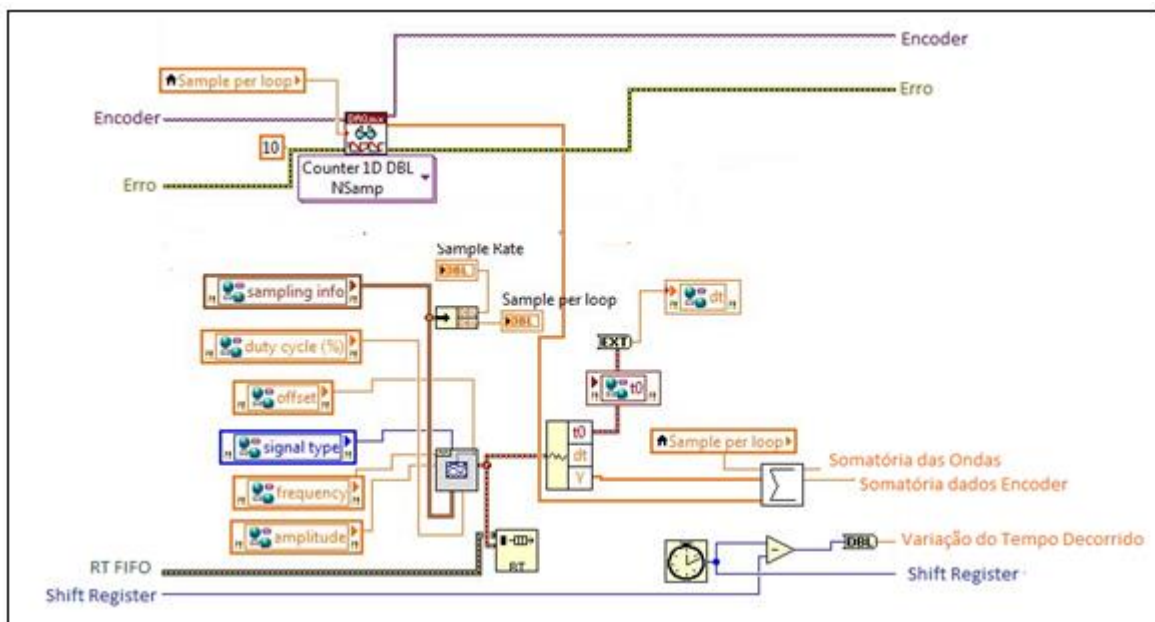


Figura 42 - Programa executado no Target parte 1

Fonte: Autoria própria.

Conforme a figura 43, se a somatória da velocidade de referência dividida pelo número de amostras, velocidade de referência média, for maior que zero o motor gira no sentido horário, contudo, caso essa divisão for menor que zero o motor inverte a rotação. O subVI Velocidade (apêndice C) transforma a posição angular do *encoder* em velocidade, este valor é salvo no RT FIFO e enviado para o *loop* de baixa prioridade. O bloco PID recebe o valor absoluto da velocidade real e da velocidade

referência e com os ganhos inseridos pelo usuário realiza os ajustes, resultando em uma ação de controle. O subVI Amostra (apêndice C) transforma o número de amostra e taxa de amostragem em tempo do *loop* e número de amostras por segundos.

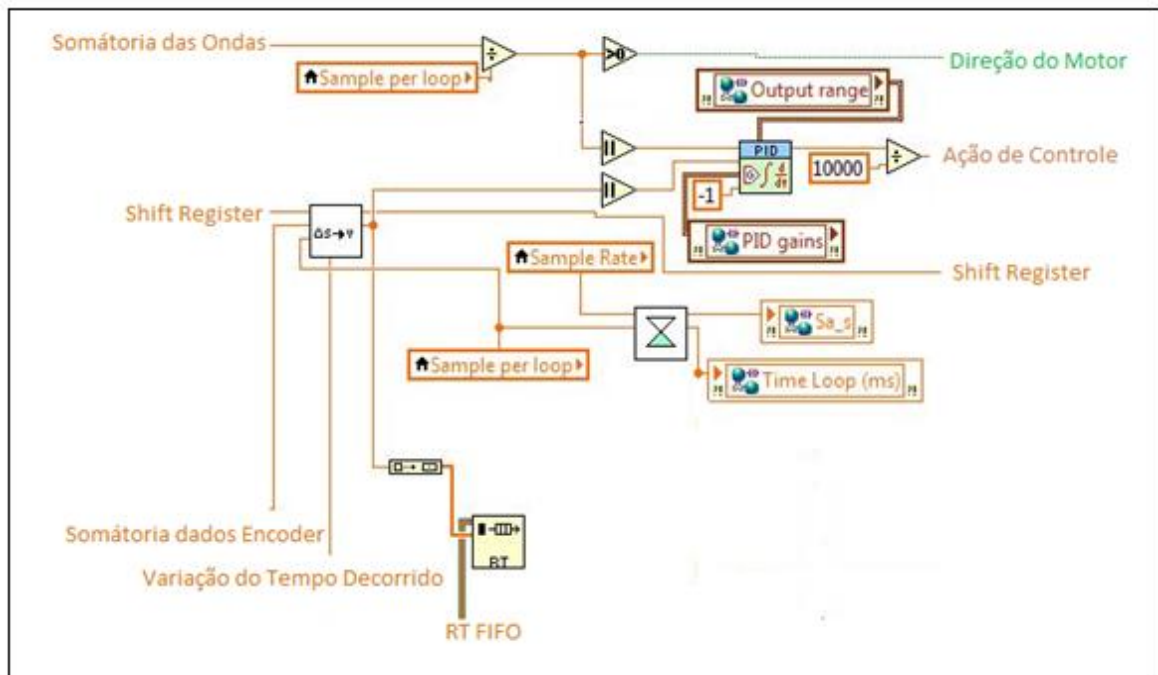


Figura 43 - Programa executado no Target parte 2

Fonte: Autoria própria.

Na figura 44, o subVI Direção (apêndice C) aciona o canal da direção do motor, se for sentido horário o IN1 é acionado e IN2 é zero, para o sentido inverso IN2 é acionado e IN1 é zero. A ação de controle, resultante do bloco PID, é salva no RT FIFO e enviada para o *loop* de baixa prioridade. A ação de controle varia de 0 a 1, quando esta assume o valor 0 ou 1 ocasiona um erro no programa de aquisição, para evitar este erro, o subVI Corrigir (apêndice D) transforma o valor 0 e o valor 1 em valores próximos. Após o tratamento, a ação de controle é enviada para o canal PWM.

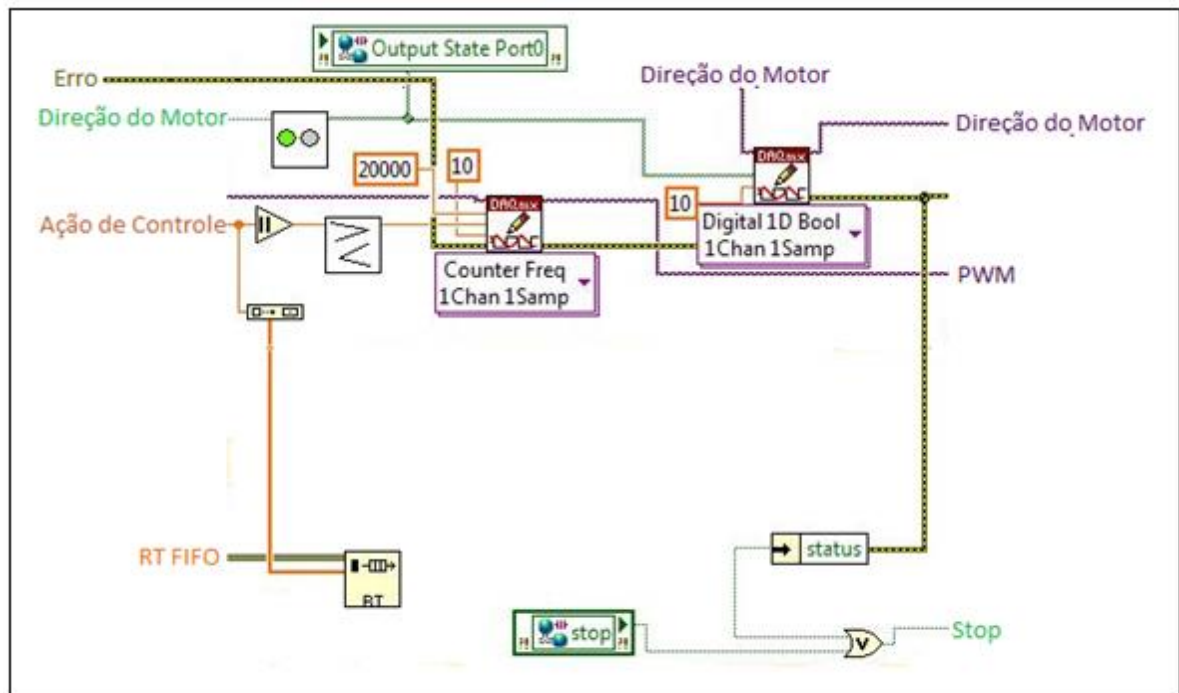


Figura 44 - Programa executado no Target parte 3

Fonte: Autoria própria.

O envio das variáveis para o host é feita no *loop* de baixa prioridade, com isso os dados da aquisição do *loop* de tempo crítico são enviados pelo RT FIFO, com um buffer de 250 amostras, para o *loop* de baixa prioridade conforme figura 45.

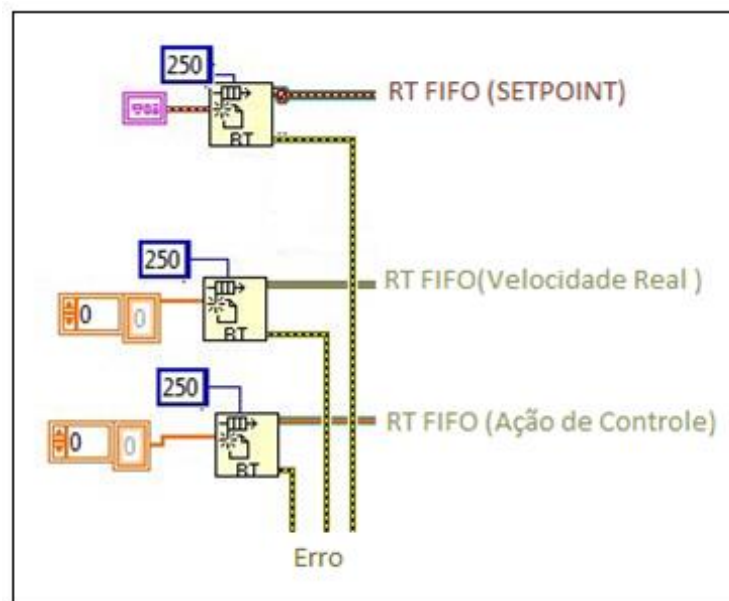


Figura 45 - Envio de dados utilizando RT FIFO

Fonte: Autoria própria.

Na figura 46, os dados são transferidos pela RT FIFO e escrito no bloco de fluxo de rede no *loop* de baixa prioridade. Com isso os dados são enviados do *target* para o *host*, contanto que as configurações dos pontos finais e endereço de IP estejam corretas.

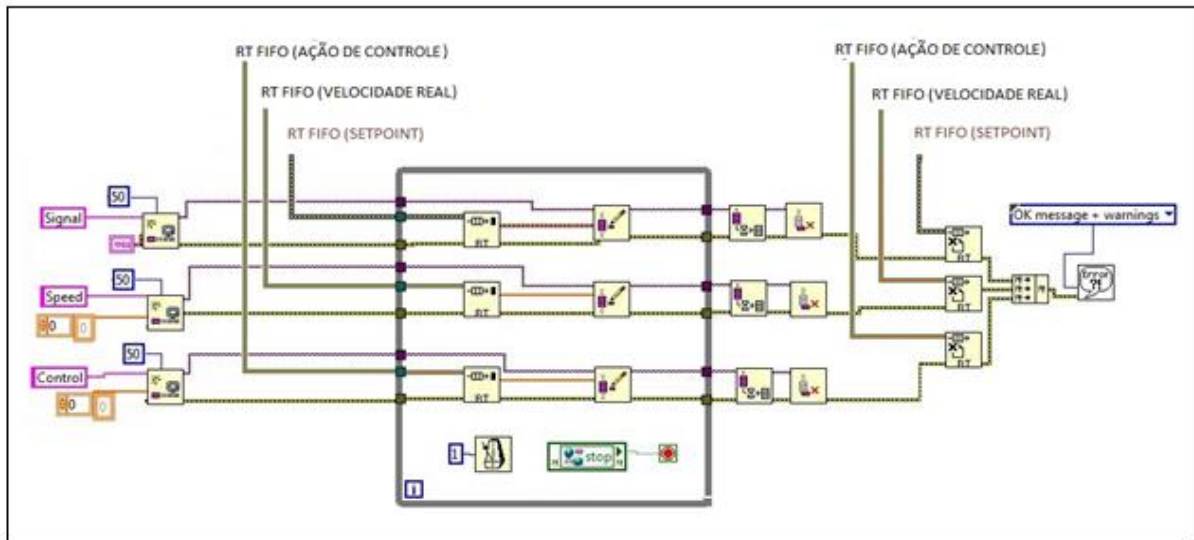


Figura 46 - Envio de dados para o host utilizando fluxo de rede

Fonte: Autoria própria.

3.2.3 Placa de aquisição

A aquisição e o controle são feitos através da placa PCI-6221, da National Instrument. Esta placa (figura 47) possui 16 entradas analógicas com 16 bits de resolução, e capacidade de aquisição de até 250KS/s.



Figura 47 - Placa PCI-6221

Fonte: Autoria própria.

A PCI-6221 é uma placa de aquisição de dados de baixo custo da série M, otimizada para aplicações nas quais o custo é uma consideração importante, podendo ser utilizada em um sistema operacional de tempo real.

A comunicação desta com o sistema físico é dada através de um cabo conectado a um conector de blocos com 37 pinos. A configuração de cada pino do conector de bloco pode ser visto na figura 48.

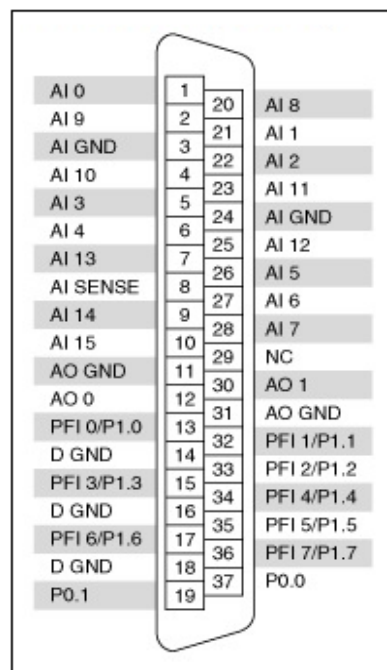


Figura 48 - Pinos da placa de aquisição

Fonte: National Instrument (2015).

A placa possui 2 contadores de 80MHz, no quadro 2 os pinos de cada contador e suas funções são apresentadas.

Contador	Número Pino	Nome do sinal
CTR 0 SCR	13	PFI 0
CTR 0 GATE	32	PFI 1
CTR 0 AUX	33	PFI 2
CTR 0 OUT	17	PFI 6
CTR 0 A	13	PFI 0
CTR 0 Z	32	PFI 1
CTR 0 B	33	PFI 2
CTR 1 SCR	15	PFI 3
CTR 1 GATE	34	PFI 4
CTR 1 AUX	35	PFI 5
CTR 1 OUT	36	PFI 7
CTR 1 A	15	PFI 3
CTR 1 Z	34	PFI 4
CTR 1 B	35	PFI 5
FREQ OUT	35	PFI 5

Quadro 2 – Pinos dos contadores

Fonte: National Instrument (2015).

Conforme visto anteriormente os sinais do PWM, IN1 e IN2 do drive de potência e os canais A e B da placa de decodificação de quadratura são enviados para a placa de aquisição. Com isto foram utilizados dois contadores, sendo o CTR 0 OUT o sinal do PWM e o CTR 1 A e B para o sinal dos canais A e B do *encoder*. Os pinos de direção do motor foram conectados as portas digitais P0.0 e P0.1, conforme figura 49.

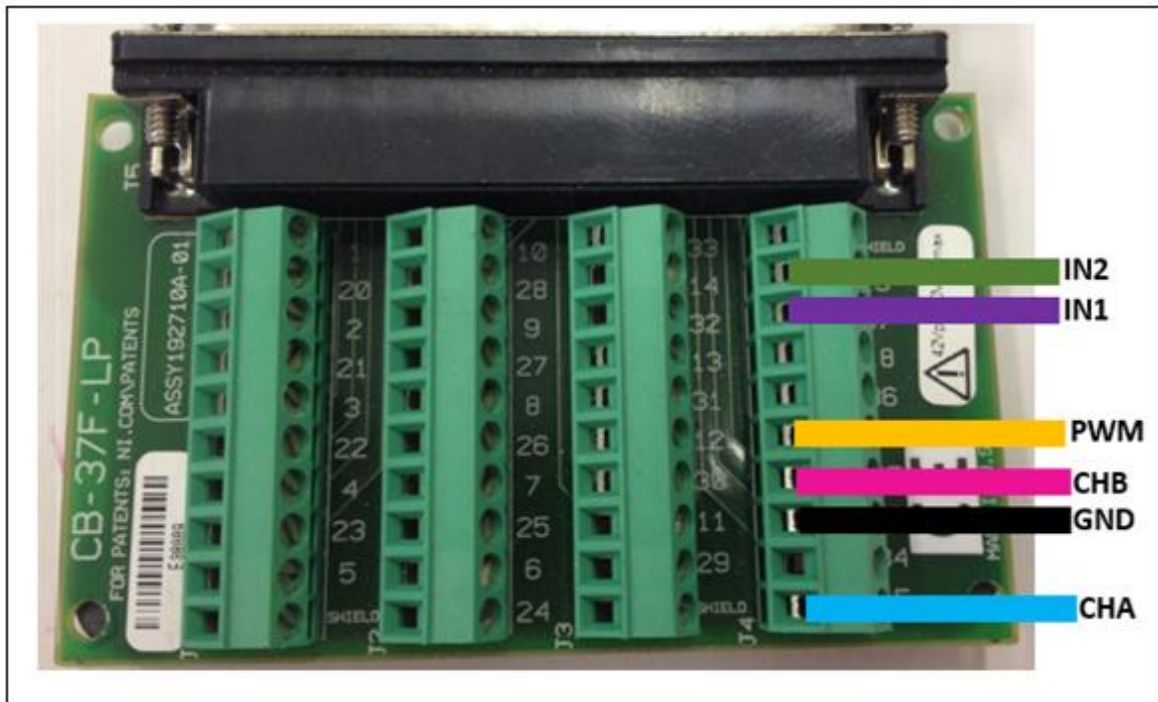


Figura 49 - Ligação dos pinos da placa de aquisição

Fonte: Autoria própria.

3.3 COMUNICAÇÃO DO SISTEMA

A comunicação entre os computadores é feita através de uma rede de conexão local Ethernet, configurando os endereços de IP, do *target* e do *host*, para a mesma rede. Neste caso o *target* possui um endereço de IP e o *host* outro como ilustra a figura 50.

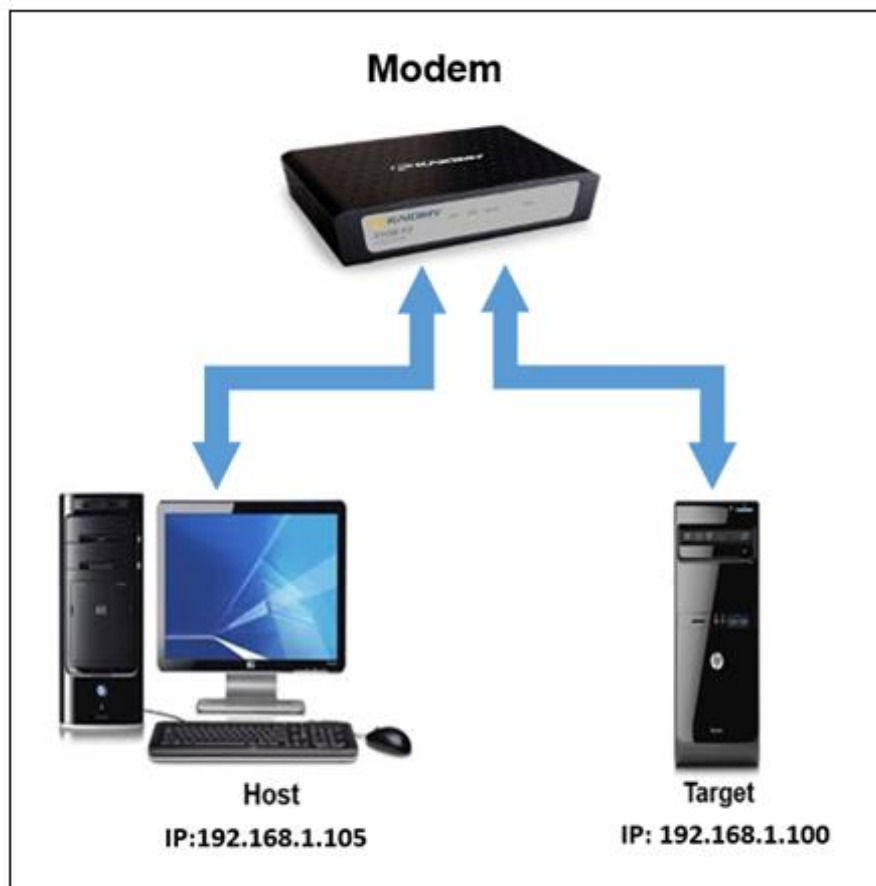


Figura 50 - Comunicação Ethernet

Fonte: Autoria própria.

Através da rede ethernet, os dados da aquisição e do controle são enviados de um computador para o outro. A figura 51 ilustra a comunicação dos dados entre os sistemas. As variáveis de entrada definidas pelo usuário (*setpoint*, parâmetros do controle e a taxa de discretização do sistema) são enviadas para o *target*. O *target* recebe e processa estes dados e envia para o servomotor o valor do PWM e a direção de rotação, o servomotor, por sua vez, envia a posição angular do motor para o *target*, onde está o controle PID sendo executado em tempo real, formando um sistema de controle em malha fechada. Os dados de velocidade, ação de controle e o tempo são enviados ao *host* para serem salvos e analisados posteriormente ou servirem para geração dos gráficos desejados.



Figura 51 - Comunicação de dados do sistema

Fonte: Autoria própria.

4 RESULTADOS

Com base na compreensão dos conceitos relacionados com controle clássico, foram realizados testes para o ajuste dos ganhos que possibilitassem um desempenho satisfatório do sistema.

Um método empírico foi utilizado na sintonia dos ganhos, apesar de pouco científico, para esta aplicação apresentou uma sintonia satisfatória. Este método consiste nas seguintes etapas:

1º) Aumentar o ganho proporcional, zerando os ganhos integral e derivativo, até a saída responder as mudanças de *setpoint*.

2º) Inserir o ganho integral.

3º) Após definidos os ganhos proporcional e integral, o ganho derivativo pode ser utilizado.

Os ajustes dos ganhos, ou *tunning*, foi realizado com uma velocidade baseada na curva de uma onda quadrada com amplitude de 1500 rpm, offset de 3000 rpm, frequência de 0,5 Hz e um *duty cycle* de 50%

Sintonizando apenas o ganho proporcional obteve o gráfico da figura 52, o sistema começou a responder as mudanças de *setpoint* com um ganho de 1,6 porem sua curva estava longe do valor desejado. Aumentando a ação proporcional obteve uma resposta mais rápida e próxima ao valor de velocidade desejado.

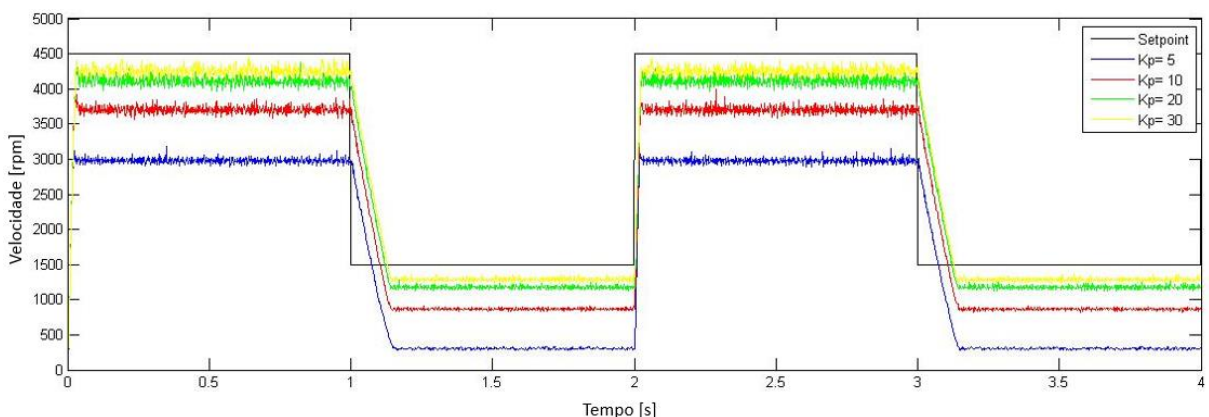


Figura 52 - Variação do ganho proporcional.

Fonte: Autoria própria.

Com o ganho proporcional respondendo as mudanças do *setpoint*, acrescentou o ganho tempo integral $T_i = 0,001$ (figura 53). Após inserir a ação de controle integral,

observou-se que quanto menor o valor de K_p maior o *undershoot*, o valor que a variável controlada ultrapassa o valor final, do sistema. Entre as curvas da figura 53, a que apresentou uma melhor resposta foi com um ganho proporcional igual a 10.

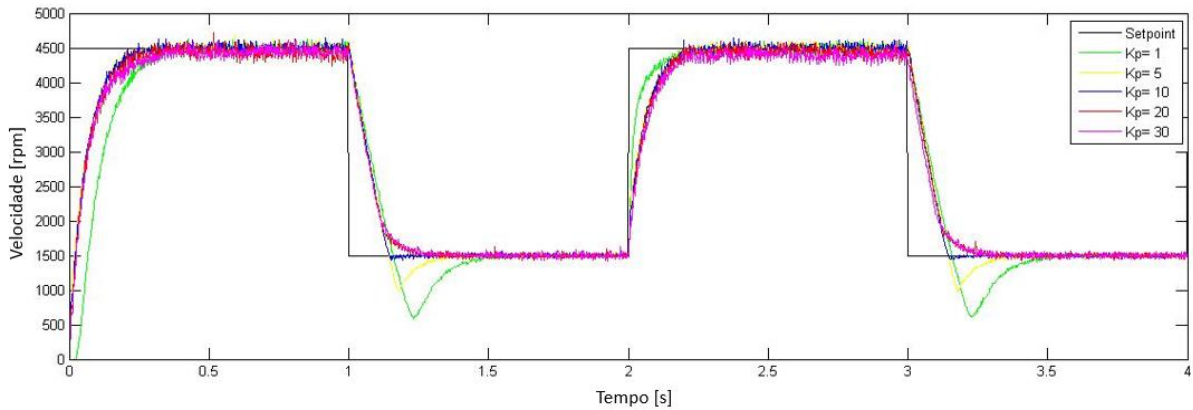


Figura 53 - Variação do ganho proporcional com $T_i = 0,001$

Fonte: Autoria própria.

Porem quanto maior o ganho proporcional maior a oscilação do sistema. A ação de controle para $K_p = 10$ e $T_i = 0,001$ apresenta esta oscilação como pode ser visto na figura 54.

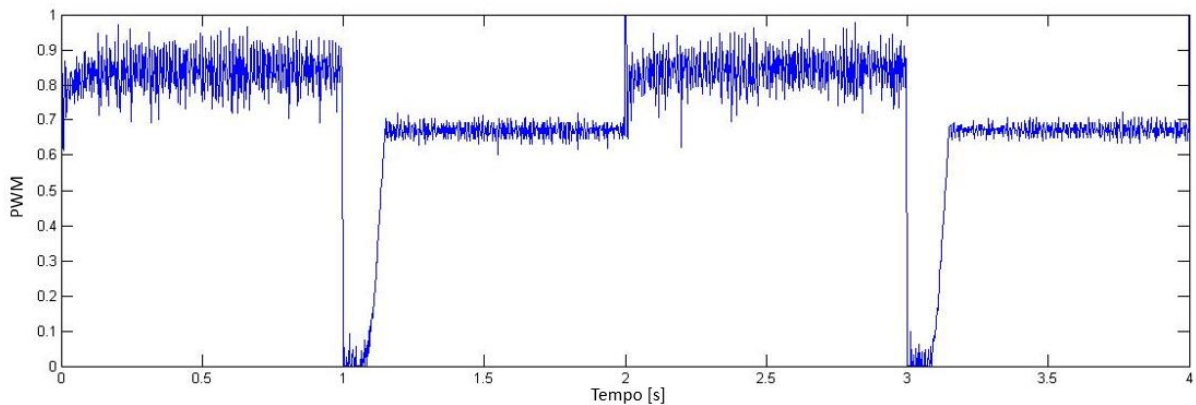


Figura 54 - Ação de controle para $K_p = 10$ e $T_i = 0,001$

Fonte: Autoria própria.

Como é indesejável grandes variações, preferiu utilizar um ganho proporcional mais baixo. A figura 55 apresenta a ação de controle para $K_p = 1$ e $T_i = 0,001$, com poucas variações quando comparada a figura 54.

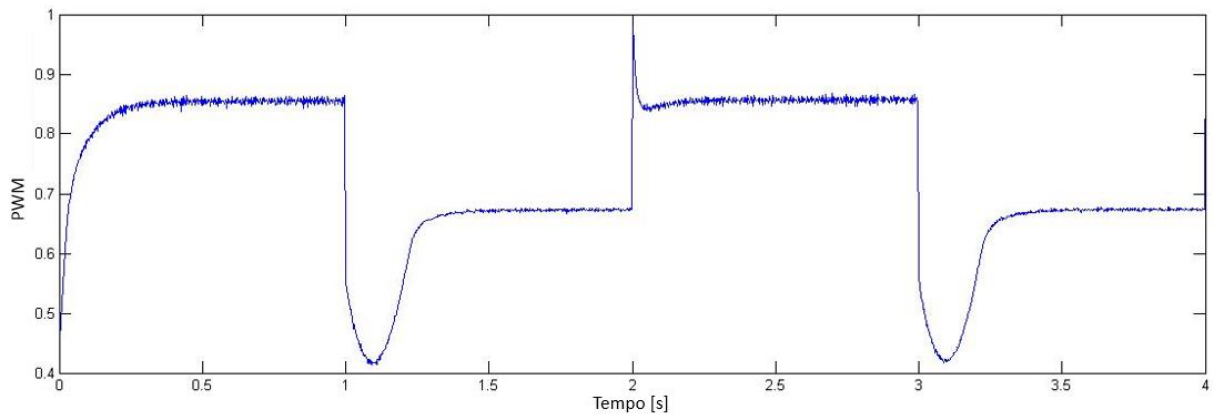


Figura 55 - Ação de controle para $K_p = 1$ e $T_i = 0,001$

Fonte: Autoria própria.

Com um ganho proporcional $K_p = 1$ e variando a ação de controle integral, obteve o gráfico da figura 56, onde o melhor ajuste da curva esta entre os valores $T_i = 0,002$ e $T_i = 0,0025$.

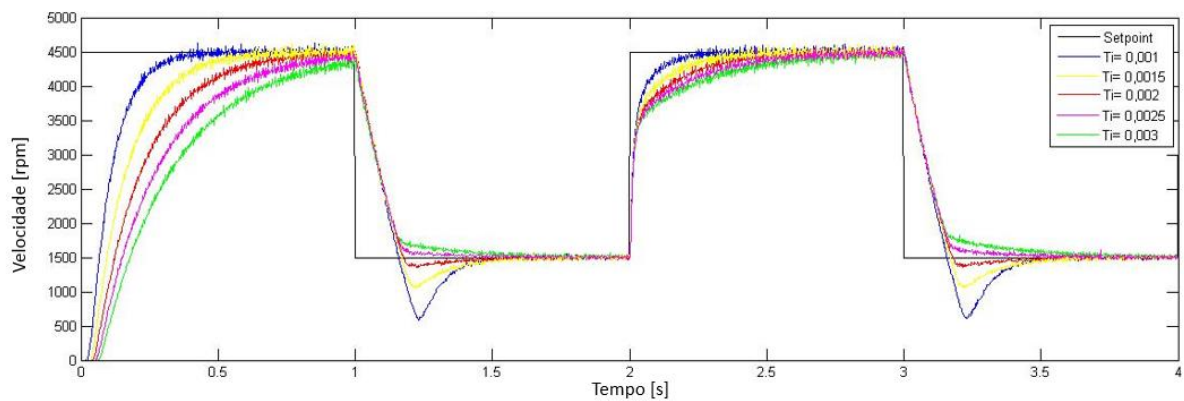


Figura 56 - Variação do ganho tempo integral com $K_p = 1$

Fonte: Autoria própria.

Através de testes variando os ganhos integrais e proporcional encontrou as correções para um desempenho satisfatório do sistema, o melhor ajuste dos ganhos foi para o valor de $K_p = 0,9$ e $T_i = 0,002$ conforme figura 57.

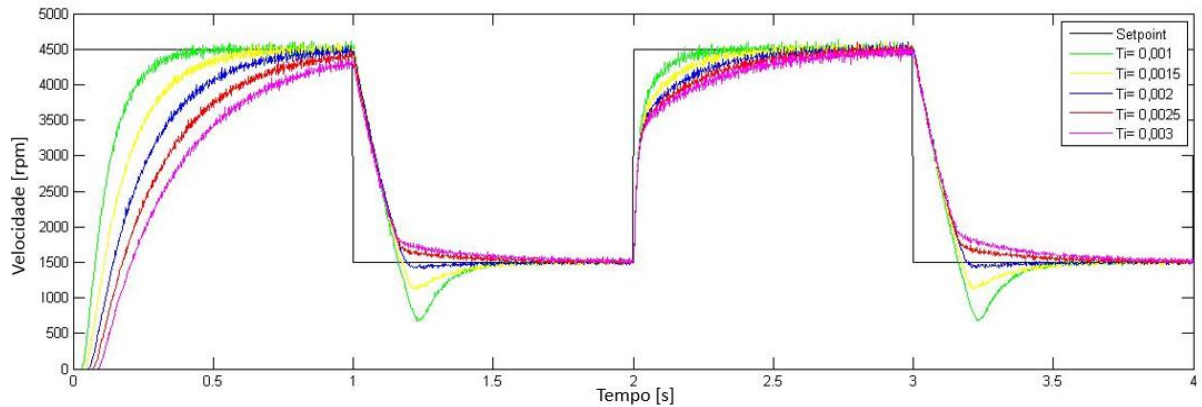


Figura 57 - Variação do ganho tempo integral com $K_p = 0,9$

Fonte: Autoria própria.

O ganho tempo derivativo não foi utilizado, pois apresentou grandes variações de velocidade. Na figura 58 este ganho foi ajustado com um valor de $T_d = 0,001$, devido a sua sensibilidade a ruídos adotou um valor baixo. O desempenho do sistema mostrou-se aceitável somente quando seu valor tendia a zero.

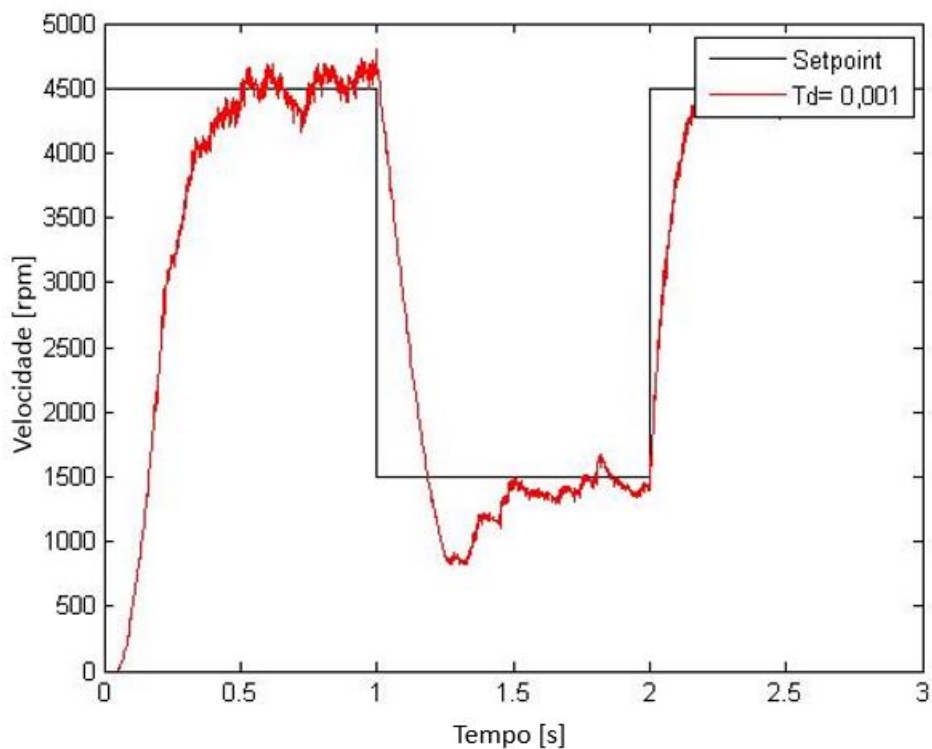


Figura 58 - Variação do ganho tempo derivativo com $K_p = 0,9$ e $T_i = 0,001$

Fonte: Autoria própria.

O método de tentativa e erro utilizado neste trabalho é um dos métodos de ajuste deste controlador, pois ainda não existe um método único para a sintonia dos ganhos. O bloco PID do LabVIEW possui uma função *Autotunnig*, onde os valores dos ganhos K_p , T_i e T_d são automaticamente obtidos de acordo com o desvio do sistema. Os valores encontrados com esta função foram $K_p = 2,192$, $T_i = 0,006$ e $T_d = 0$. Inserindo os ganhos encontrados pelo *Autotunnig*, observou através da figura 59, que o sistema não atinge o valor desejado de 4500 rpm, e comparando com o ajuste encontrado anteriormente os parâmetros encontrados através desta função não foram satisfatório.

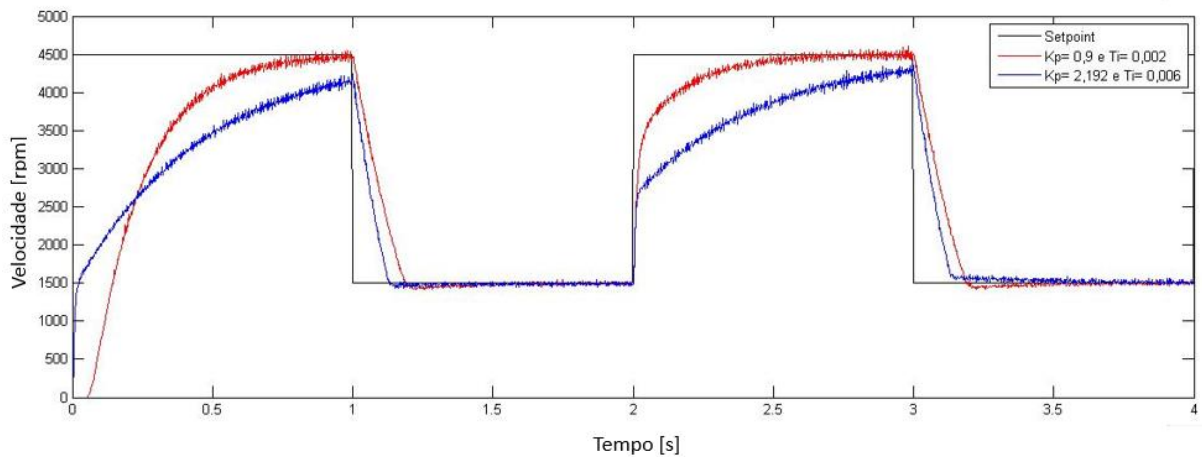


Figura 59 - Comparação dos ganhos da função Autotunnig

Fonte: A autoria própria.

Com isso os valores dos ganhos para o controle da velocidade do servomotor foram $K_p = 0,9$, $T_i = 0,001$ e $T_d = 0$. Para demonstrar o funcionamento do sistema de controle foram realizados teste com diferentes *setpoints*, utilizando os parâmetros do programa, como variação da amplitude, do offset, frequência e formas de ondas.

Utilizando uma amplitude de 1500 rpm, um offset de 1000 rpm e uma frequência de 0,5 Hz, obteve-se uma velocidade negativa, ou seja, houve a inversão da rotação do servomotor. O gráfico da figura 60 apresentou um *undershoot*, isto ocorre pois o motor para de girar no sentido horário e começa a girar no sentido anti-horário demorando um tempo maior para alcançar a velocidade estabelecida.

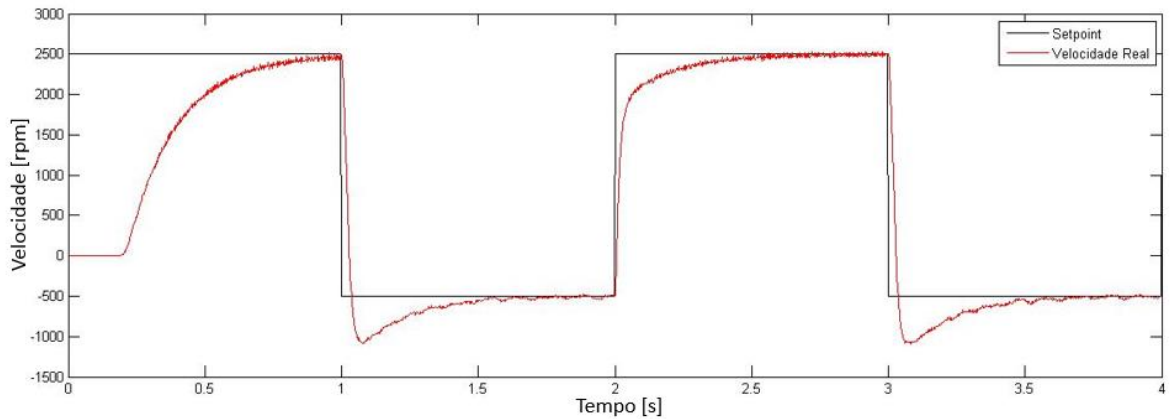


Figura 60 - Resultado da velocidade do servomotor com inversão de rotação.

Fonte: Autoria própria.

Para variar a velocidade do motor, foi utilizado uma frequência de 0,5 Hz, amplitude de 2000 rpm e uma variação no offset. Conforme figura 61, o motor apresentou um limite de 7000 rpm, por isso o controle não respondeu para velocidade acima deste limite.

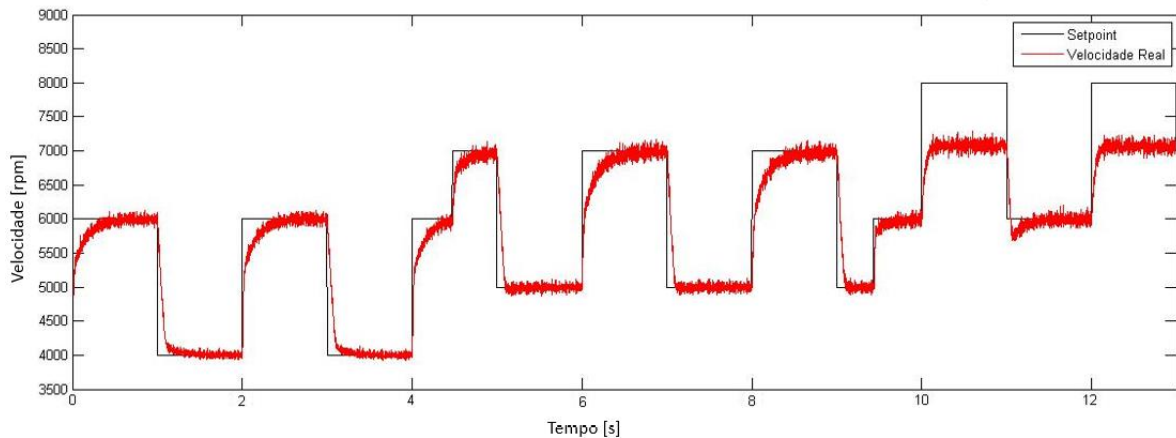


Figura 61- Resultado da variação da velocidade do servomotor

Fonte: Autoria própria.

Adicionando uma carga mecânica (figura 62) limitou-se a velocidade do motor para 6000 rpm. Porém, apresentou um controle satisfatório para velocidades mais reduzidas.

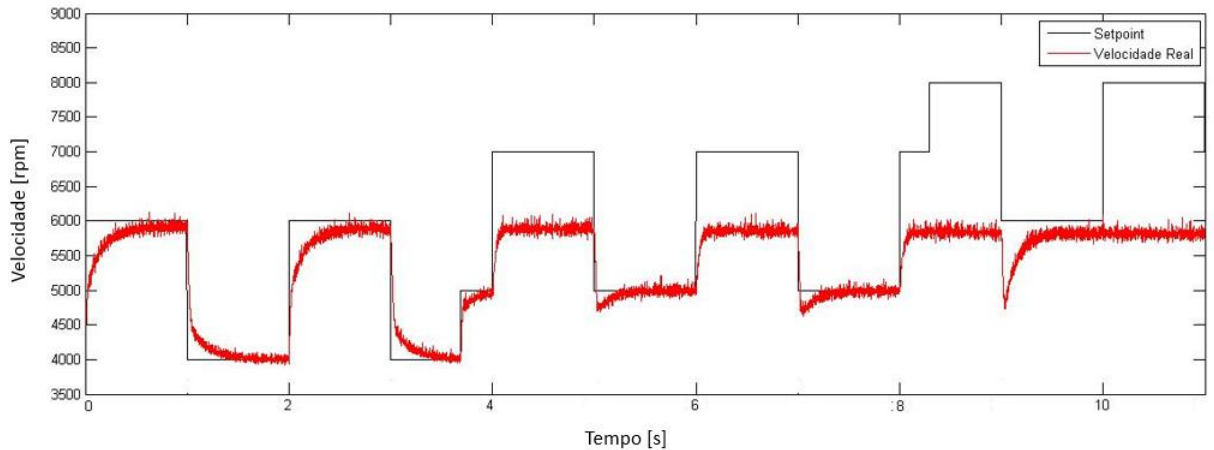


Figura 62 - Resultado da variação da velocidade com carga mecânica

Fonte: Autoria própria.

Com uma amplitude de 1500 rpm, offset de 3000 rpm, frequência de 0,1 Hz e variando as formas de onda, apresentou um controle satisfatório para os três tipos de ondas quadrada, triangular e senoidal, como pode ser visto na figura 63.

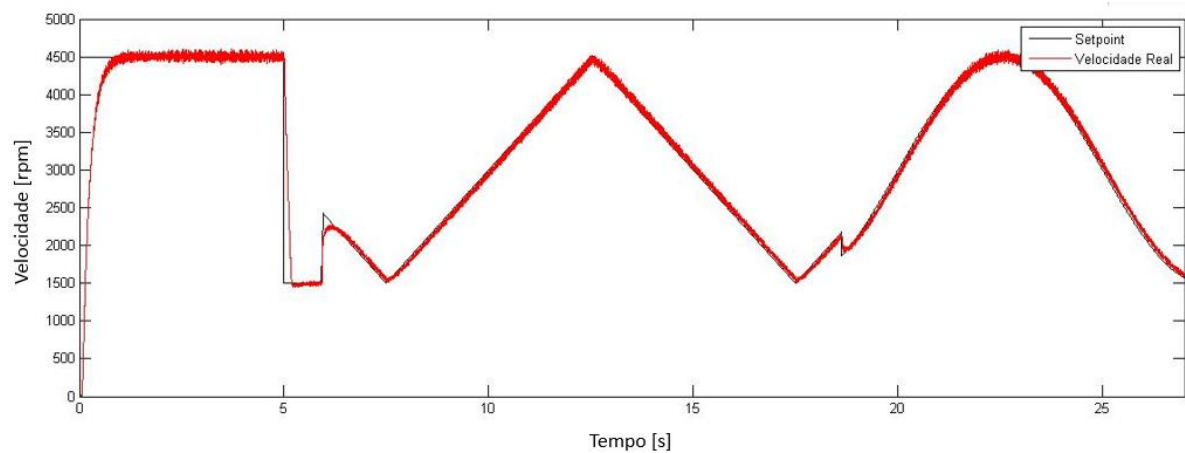


Figura 63 - Resultado da variação dos tipos de onda.

Fonte: Autoria própria.

Conforme a figura 64, um distúrbio no eixo do servomotor, em aproximadamente 3s, 7s, 8s, 10s, 15s e em 19s, foi aplicado para analisar o comportamento do controle nesta situação. Pôde-se perceber que as perturbações foram corrigidas rapidamente.

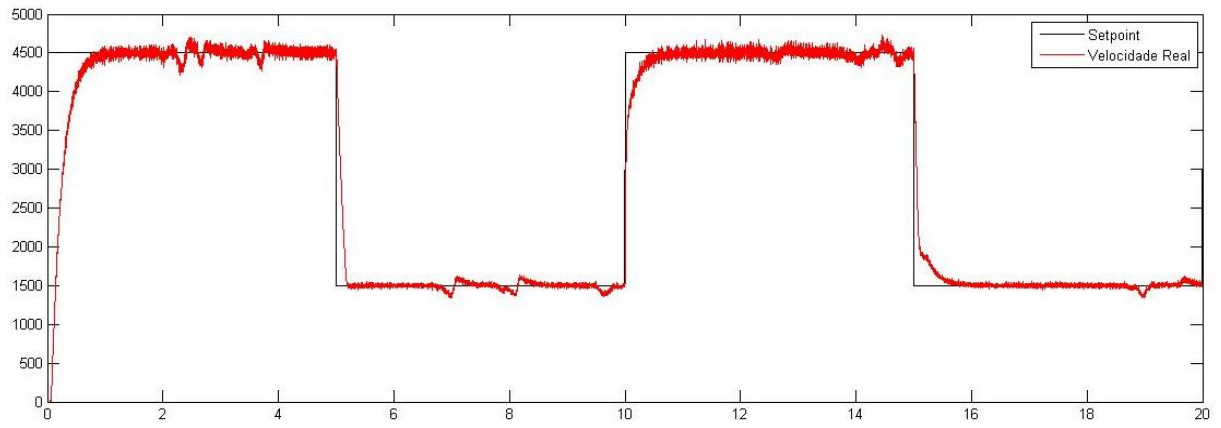


Figura 64 - Resultado da velocidade com distúrbios no eixo do servomotor

Fonte: Autoria própria.

5 CONCLUSÃO

Este trabalho teve como foco central o desenvolvimento de um kit didático para controle de velocidade de um servomotor, com aquisição de dados em tempo real, utilizando um sistema de controle baseado nas ações proporcional, integral e derivativa. Com o auxílio do *software* LAbVIEW foi possível a criação do sistema de controle e aquisição, além de possibilitar o desenvolvimento de uma interface intuitiva para o usuário.

Encontrar o melhor ajuste das ações de controle PID neste trabalho é importante, pois através dos ganhos é possível obter uma velocidade próxima ao valor desejado e com isso analisar se o sistema de controle funciona corretamente. Em virtude dos resultados encontrados para os ganhos, e com a análise deste com as diversas funções apresentadas pode-se fazer a validação do sistema de controle, pois este atende a proposta de realização da correção do erro encontrado pelo sistema.

A interface para o usuário foi desenvolvida para facilitar a manipulação das variáveis e possibilitando, através dos gráficos, o usuário observar a resposta do sistema em tempo real, outra opção da interface é o armazenar os dados com isso a taxa de aquisição e os dados para determinados tempos podem ser analisados posteriormente. A interface mostrou-se flexível e com uma interação com todas as variáveis envolvidas no processo em tempo real podendo ser utilizada como ferramenta de ensino contribuindo para o aprimoramento de conhecimentos teóricos referentes a sistemas de controle. Através dos conhecimentos obtidos nas aulas teóricas o aluno consegue sintonizar os ganhos e analisar a influência de cada um em um sistema de controle. Com a modelagem do motor é possível aplicar ferramentas de controle clássico como lugar das raízes e diagrama de Bode e analisar a eficácia dessas ferramentas na prática.

Como trabalhos futuros, sugere-se projetar outros sistemas de controle utilizando outras técnicas de controle. Além da obtenção de um modelo matemático validado para o motor que possibilitaria a comparação entre os sistemas simulados e os experimentais que é o objetivo recorrente em sistemas de controle.

REFERÊNCIAS

ALECRIM, Emerson. **Endereço IP (Internet Protocol)**. Disponível em <<http://www.infowester.com/ip.php>>. Acesso em: 05 set. 2015.

ANDRADE, Hudson. **Sensor de Velocidade**. Disponível em <http://www.dee.ufrn.br/~luciano/arquivos/ins_ele/Apresenta%E7%F5es_2008_2/HUDSON%20PINHEIRO/Sensor_de_velocidade.pdf>. Acesso em: 04 set. 2015.

BARCESSAT, Jaime; TABARANA, Thiago. **Controle Digital de um servo-posicionador utilizando o programa LabVIEW**. 2006. 59 f. Trabalho de Conclusão de Curso (Engenharia da Computação) - Instituto de Estudos Superiores da Amazônia, 2006. Disponível em <<http://www3.iesam-pa.edu.br/ojs/index.php/computacao/article/viewFile/89/84>>. Acesso em: 14 set. 2015.

BATISTA, Manuel. **Sistemas de Aquisição de Dados**. Disponível em <http://www.estgv.ipv.pt/paginaspessoais/maeb/im/Teorica_Bibliografia/Cap_E_Sistemas%20de%20Aquisi%C3%A7%C3%A3o%20de%20Dados/1-Introdu%C3%A7%C3%A3o/Texto%20de%20Estudo%20-%20Sistemas%20de%20Instrumenta%C3%A7%C3%A3o%20-%20Capitulo%204.pdf>. Acesso em: 04 set. 2015.

BOK, J.; SCHAUER, P. **LabVIEW-based control and data acquisition system for cathodoluminescence experiments**. Review of Scientific Instruments, v. 82, 2011.

BRANDINO, Wandreson. **Apostila TCP/IP**. Disponível em <http://www.gamaesouza.edu.br/site/ENADE/Redes/TCP_IP.pdf>. Acesso em: 08 set. 2015.

COETZEE, BRETT. **Data Communication in LabVIEW**. Disponível em <<http://southafrica.ni.com/sites/default/files/Data%20Communication%20in%20LabVIEW.pdf>>. Acesso em: 23 set. 2015.

COSTA, Cesar. **Software LabVIEW**. Disponível em <http://professorcesarcosta.com.br/upload/imagens_upload/Aulas_Pr%C3%A1ticas_Apostila%20LabView_Prof.pdf>. Acesso em :02 set. 2015.

ELLIOTT, C.; VIJAYAKUMAR, V.; ZINK, W.; HASEN, R. **National Instruments LabVIEW: A Programming Environment for Laboratory Automation and Measurement.** Journal of the Association for Laboratory Automation, v.12, p. 17–24, 2007

GARCIA, R. F.; QUEIROZ, D. M.; MIYAGAKI, O. H.; PINTO, F. D. **Programa computacional para aquisição de dados para avaliação de máquinas agrícolas.** Revista Brasileira de Engenharia Agrícola E Ambiental, v.7, p. 375–381, 2003.

GOUVEIA, Danilo; ANDRADE, Fabricio; BALTAZAR, Phellype. **Controle de força em sistemas robóticos e mecatrônicos.** 2011. 8 f. Proposta de Trabalho de Conclusão de Curso (Engenharia Elétrica) - Universidade Tecnológica Federal do Paraná, 2011. Disponível em <http://nupet.daelt.ct.utfpr.edu.br/tcc/engenharia/doc-equipe/2011_2_29/2011_2_29_proposta.pdf>. Acesso em: 14 set. 2015.

HALVORSEN, Petter. **Virtual Instruments with LabVIEW.** Disponível em <<http://home.hit.no/~hansha/training/daq/documents/Virtual%20Instruments%20with%20LabVIEW.pdf>>. Acesso em: 06 set. 2015.

KEHTARNAVAZ, N; LOIZOU, P; RAHMAN, M. **Introduction to LabVIEW.** Disponível em <http://cnx.org/contents/484d4446-da50-4af4-bc44-d6a2e0d660df@14.1:3/An_Interactive_Approach_to_Sig%20Introduction%20to%20LabVIEW>. Acesso em: 04 set. 2015.

MATOS, Nuno. **Análise do Funcionamento de um Servomotor de Corrente Alternada com Ímãs Permanentes.** 2012. 86 f. Dissertação (Mestrado em Engenharia Eletrotécnica e Computadores) – Universidade do Porto, 2012. Disponível em <<https://repositorio-aberto.up.pt/bitstream/10216/72684/1/000155343.pdf>>. Acesso em: 22 set. 2015.

MELLO, Carlos. **Sistemas de Controle (Servomecanismo).** Disponível em <<http://www.cin.ufpe.br/~cabm/servo/Aula00.pdf>>. Acesso em: 14 set. 2015.

MOREIRA, Albert; NEGREIROS, Thiago; PORTNOI, Marcos. **Pulse Width Modulation.** Disponível em <<https://www.eecis.udel.edu/~portnoi/academic/academic-files/pwm.html>>. Acesso em: 08 set. 2015.

NASSER, P.; MELO, Charles. **Desenvolvimento de um controlador PID digital.** IN: ASSOCIAÇÃO BRASILEIRA DE EDUCAÇÃO DE ENGENHARIA, 2014, Juiz de Fora. Anais eletrônicos ...Juiz de Fora: COBENGE, 2014. Disponível em <<http://www.abenge.org.br/cobenge-2014/Artigos/129858.pdf>>. Acesso em: 04 SET. 2015.

NATIONAL INSTRUMENT. **CompactRIO Developers Guide**. Disponível em <<http://www.ni.com/pdf/products/us/criodevguideintro.pdf>>. Acesso em: 04 set. 2015.

NATIONAL INSTRUMENT. **Conceitos básicos do ambiente LabVIEW**. Disponível em <<http://www.ni.com/getting-started/labview-basics/pt/environment>>. Acesso em: 06 set. 2015.

NATIONAL INSTRUMENT. **Designing Applications Using Network Streams**. Disponível em <<http://zone.ni.com/reference/en-XX/help/371361G-01/lvconcepts/usingstreams/>>. Acesso em: 04 set. 2015.

NATIONAL INSTRUMENT. **Instalação e configuração de dispositivos DAQ PCI/PCI Express e PXI/PXI Express**. Disponível em <<http://www.ni.com/getting-started/set-up-hardware/data-acquisition/pt/pci-pxi>>. Acesso em: 10 set. 2015.

NATIONAL INSTRUMENT. **Lossless Communication with Network Streams: Components, Architecture, and Performance**. Disponível em <<http://www.ni.com/white-paper/12267/en/>>. Acesso em: 23 set. 2015

NATIONAL INSTRUMENT. **Medições com encoders de quadratura: Guia prático**. Disponível em <<http://www.ni.com/tutorial/7109/pt/>>. Acesso em: 10 set. 2015.

NATIONAL INSTRUMENT. **O que é a Tecnologia de Tempo Real?** Disponível em <<http://www.ni.com/white-paper/3938/pt/>>. Acesso em: 10 set. 2015.

NATIONAL INSTRUMENT. **Publishing Latest Values with Shared Variables**. Disponível em <http://zone.ni.com/reference/en-XX/help/371361L-01/lvconcepts/sv_publishing_latest_values/>. Acesso em: 04 set. 2015.

NATIONAL INSTRUMENT. **Sharing Data Locally on an RT Target (Real-Time Module)**. Disponível em <http://zone.ni.com/reference/en-XX/help/370622K-01/lvrtconcepts/passing_data_between_vis/>. Acesso em: 05 set. 2015.

NATIONAL INSTRUMENT. **Understanding Shared Variable Technology**. Disponível em <http://zone.ni.com/reference/en-XX/help/371361L-01/lvconcepts/ni_psp/>. Acesso em: 08 set. 2015.

NATIONAL INSTRUMENT. **Using the LabVIEW Shared Variable**. Disponível em <<http://www.ni.com/white-paper/4679/en/>>. Acesso em: 23 set. 2015.

NOGUEIRA, Marcelo. **Introdução aos Sistemas de Controle**. Disponível em <http://www.noginfo.com.br/arquivos/CC_AEC_02.pdf>. Acesso em: 14 set. 2015.
PINHEIRO, Breno. **Sistema de controle tempo real embarcado para automação de manobra de estacionamento**. 2009. 115 f. Dissertação (Mestrado em Automação e Sistemas) - Universidade Federal de Santa Catarina, 2009. Disponível em <<http://docplayer.com.br/3955960-Breno-carneiro-pinheiro-sistema-de-controle-tempo-real-embarcado-para-automacao-de-manobra-de-estacionamento.html>>. Acesso em: 25 set. 2015.

POLLOCK, Matt. **Data Communication Methods for Embedded Systems**. Disponível em <<https://decibel.ni.com/content/docs/DOC-39101>>. Acesso em: 02 set. 2015.

ROCHA, Cleiton. **Técnicas de controle de temperatura utilizando LabVIEW**. 2009. 65 f. Trabalho de conclusão de curso (Engenharia de Controle e Automação) - Universidade Federal de Ouro Preto, 2009. Disponível em <<http://www.em.ufop.br/cecau/monografias/2009/Cleiton%20F.%20da%20Rocha.pdf>>. Acesso em: 25 set. 2015.

SEELAENDER, Guilherme. **Emulação e co-simulação do sistema de controle de atitude da PMM e do sistema eletro-hidráulico de uma aeronave usando FPGAS**. 2009 195 f. Dissertação (Mestrado em Mecânica Espacial e Controle) – Instituto Nacional de Pesquisa Espaciais, 2009. Disponível em <<http://mtc-m18.sid.inpe.br/col/sid.inpe.br/mtc-m18@80/2009/04.09.15.46/doc/publicacao.pdf?ibiurl.language=en>>. Acesso em: 02 set. 2015.

SILVA, Flavio. **Planta didática-controle PID digital para motor DC**. 2008. 53 f. Trabalho de conclusão de curso (Engenharia da Computação) - Universidade Positivo, 2008. Disponível em <<http://www.up.edu.br/blogs/engenharia-da-computacao/wp-content/uploads/sites/6/2015/06/2008.14.pdf>>. Acesso em: 25 set. 2015.

TELECO. **Tutoriais Banda Larga**. Disponível em <http://www.teleco.com.br/tutoriais/tutorialredeswifi2/pagina_5.asp>. Acesso em: 10 set. 2015.

TRAJANO CAMARGO. **Controle e Automação**. Disponível em <http://www.trajanocamargo.com.br/wp-content/uploads/2012/05/Controle_e_-Automacao_Industrial_II.pdf>. Acesso em: 08 set. 2015.

VIZCAINO, J; SEBASTIA, J. **LabVIEW: entorno gráfico de programación.** Disponível em <https://books.google.com.br/books?hl=pt-BR&lr=&id=ZFQua3-eeQEC&oi=fnd&pg=PA21&dq=labview&ots=qH_tO6PEgp&sig=8w0KLDplmWzbBCEsFACzEIZFOIQ#v=onepage&q=labview&f=false>. Acesso em: 04 set. 2015.

APÊNDICE B – Diagrama de Blocos do Target

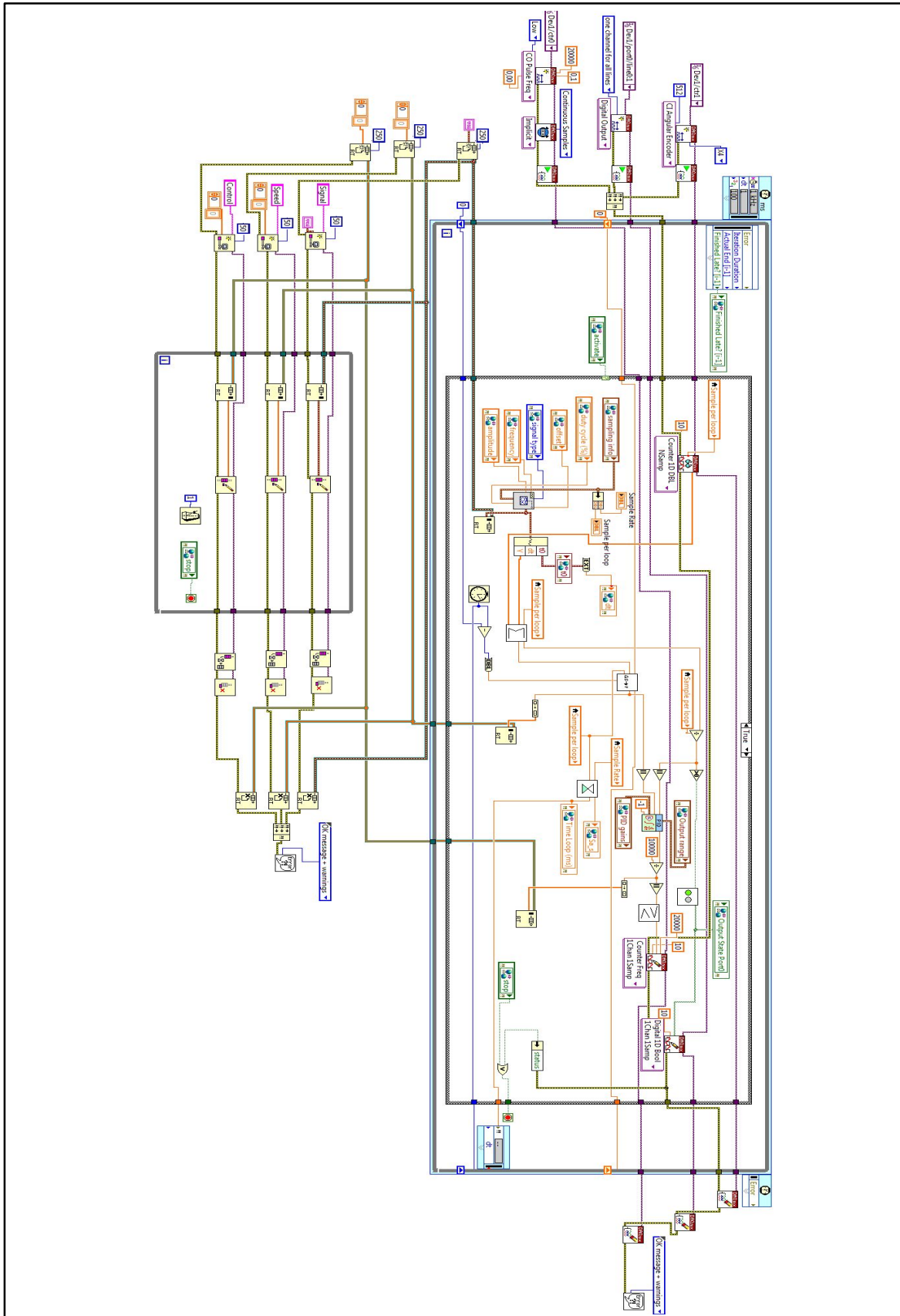


Figura 66 - Diagrama de Blocos Target

Fonte: Autoria própria.

APÊNDICE C – SUBVIS

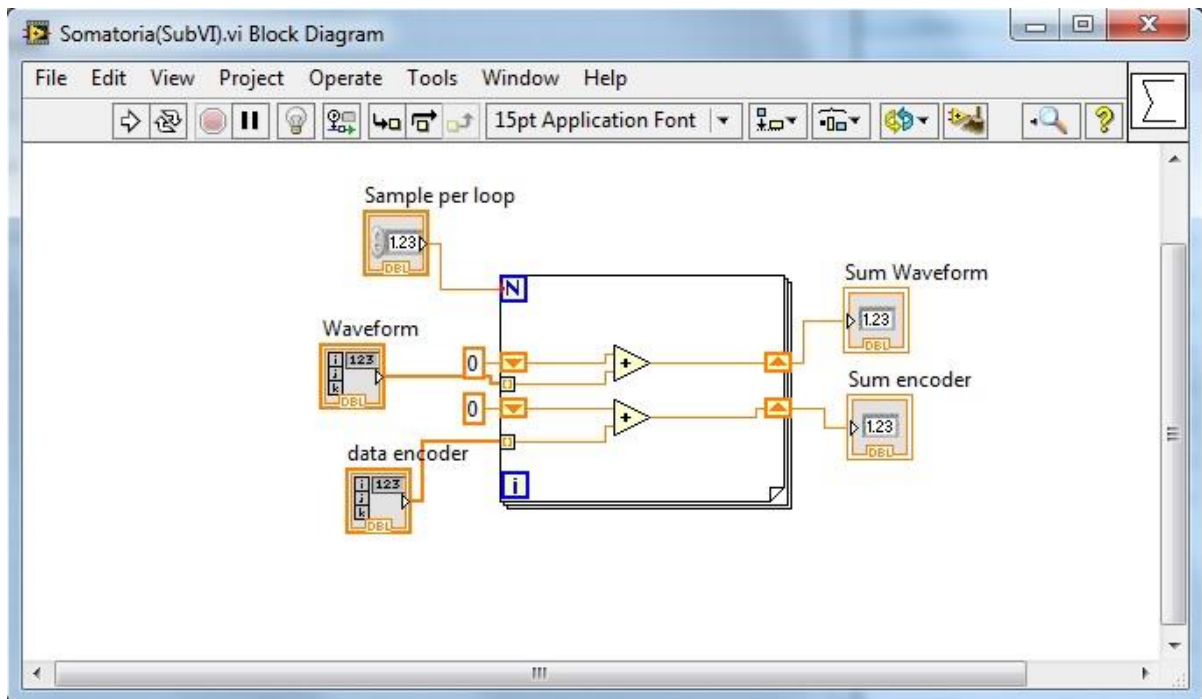


Figura 67 - SubVI Somatória

Fonte: Autoria própria.

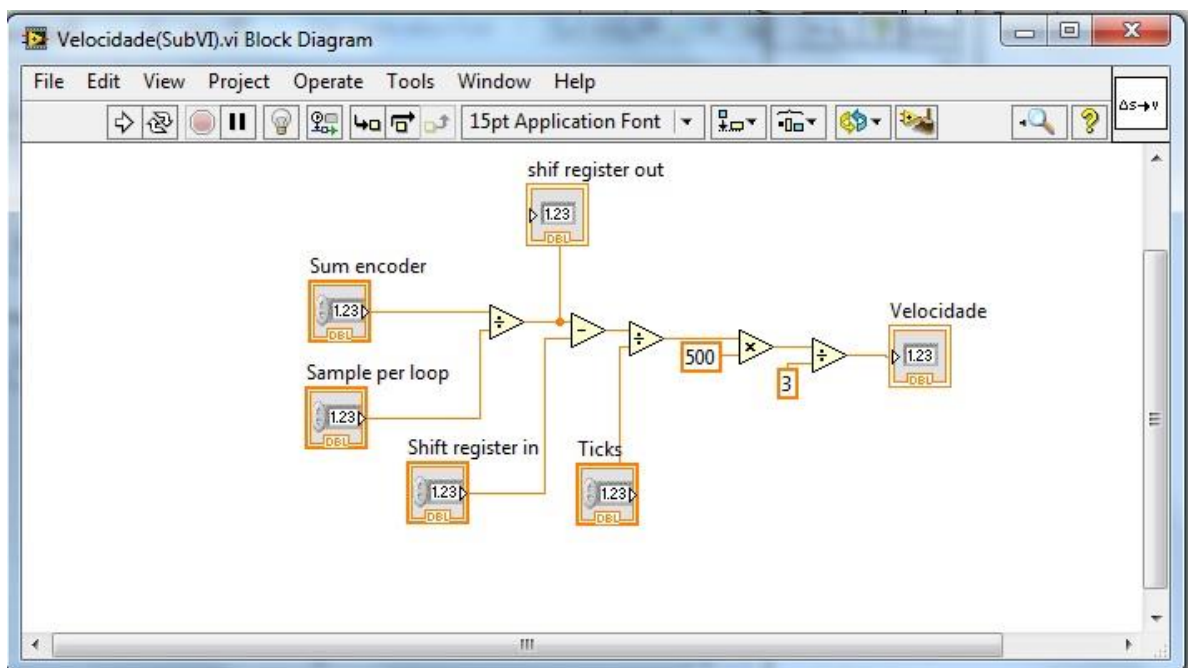


Figura 68 - SubVI Velocidade

Fonte: Autoria própria.

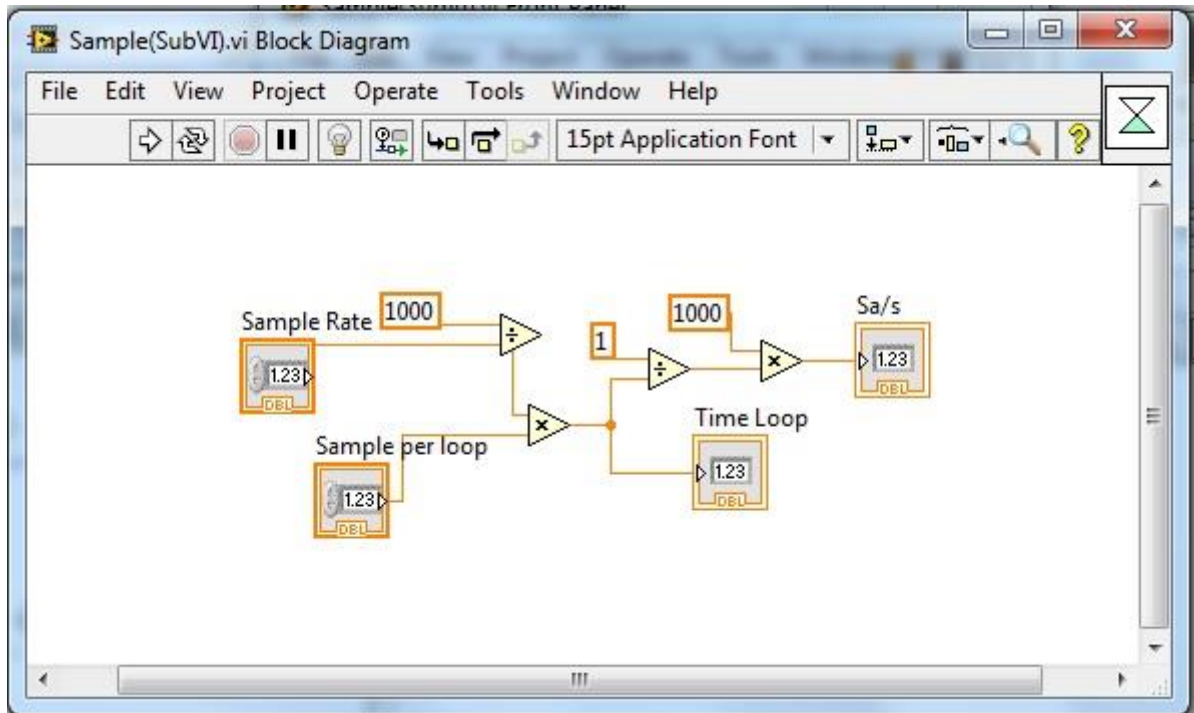


Figura 69 - SubVI Amostra

Fonte: Autoria própria.

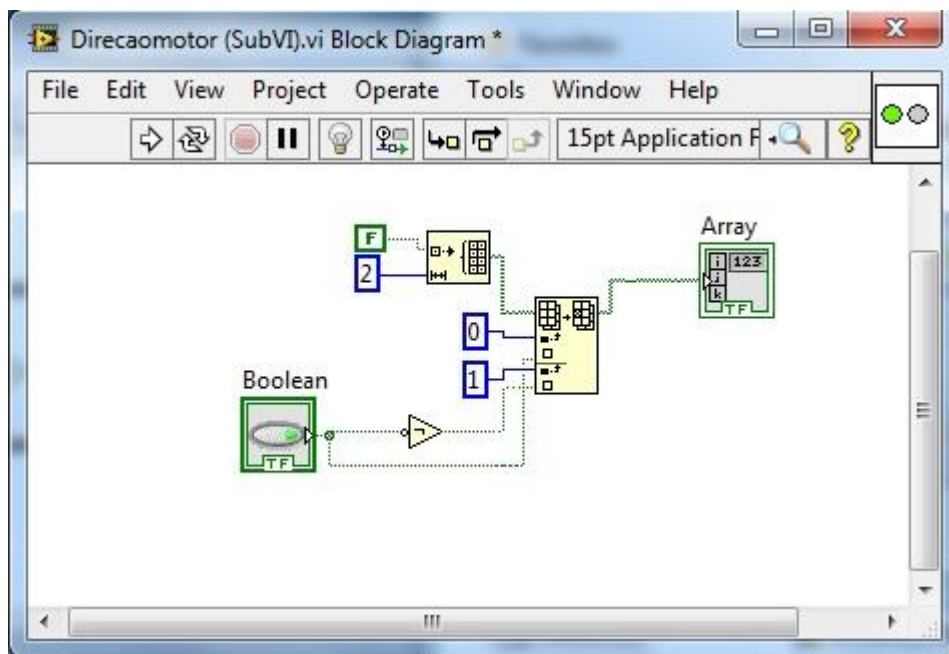


Figura 70 - SubVi Direção do motor

Fonte: Autoria própria.

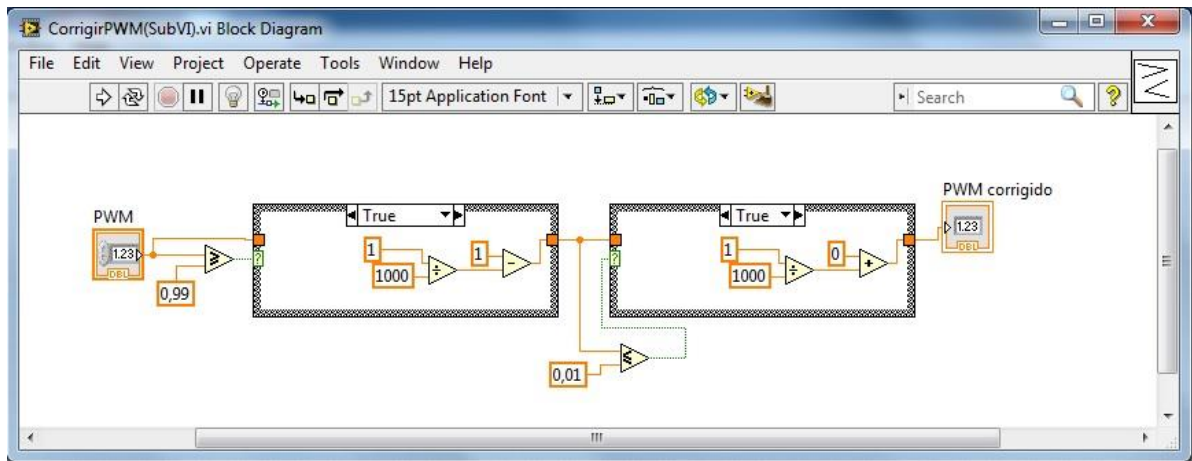


Figura 71 - SubVI Corrigir PWM

Fonte: Autoria própria.