



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Curso de Graduação em Engenharia Elétrica

CHRISTIAN CHARLES DIAS

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

Campina Grande, Paraíba

Maio de 2016

CHRISTIAN CHARLES DIAS

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

Relatório de Estágio
Supervisionado submetido à Unidade
Acadêmica de Engenharia Elétrica da
Universidade Federal de Campina Grande
como parte dos requisitos necessários
para a obtenção do grau de Bacharel em
Ciências no Domínio da Engenharia
Elétrica.

Área de Concentração: Controle e Automação

Orientador:

Professor George Acioli Júnior, Dr. Sc.

Campina Grande, Paraíba

Mai de 2016

CHRISTIAN CHARLES DIAS

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

Relatório de Estágio Supervisionado submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Controle e Automação

Aprovado em ____ / ____ / ____

Professor Dr. George Acioli Júnior
Universidade Federal de Campina Grande
Orientador

Professor Avaliador
Universidade Federal de Campina Grande
Avaliador

AGRADECIMENTO

Agradeço aos Professores Péricles Rezende Barros e George Acioli Júnior, pela oportunidade de estagiar no LIEC e pelas orientações prestadas não só durante a realização do estágio, mas durante a maior parte do curso.

Agradeço à minha família, em especial à minha mãe Maria da Conceição, por apoiar a realização desse sonho.

E por fim, a todos os colegas do Laboratório de Instrumentação Eletrônica e Controle, em especial à Rafael Lima, Lucas Moreira e Simões Toledo, cuja ajuda foi de grande importância para a realização desse trabalho.

RESUMO

Este relatório apresenta as atividades realizadas pelo aluno Christian Charles Dias durante o Estágio Supervisionado no Laboratório de Instrumentação Eletrônica e Controle (LIEC), pertencente ao Departamento de Engenharia Elétrica (DEE) da Universidade Federal de Campina Grande (UFCG), sob orientação do Professor George Acioli Júnior e supervisão do Professor Péricles Rezende Barros.

Palavras-chave: Plataforma XY, mesa coordenada, Arduino, MATLAB GUI, controlador PID.

LISTA DE FIGURAS

Figura 1 – Fachada do LIEC.....	3
Figura 2 - Planta didática para identificação e controle de um posicionador de dois eixos. ...	6
Figura 3 – Diagrama com os principais componentes da planta didática.....	6
Figura 4 – Interface da planta didática.....	7
Figura 5 - Partes que compõem o eixo de deslocamento.	8
Figura 6 - 75:1 Metal Gearmotor 25Dx54L mm LP 6V with 48 CPR Encoder.	9
Figura 7 - Dual VNH2SP30 Motor Driver Carrier MD03A, a) Vista do módulo montado e b) Circuito eletrônico do módulo.	12
Figura 8 – Layout da PCB para expensor.....	13
Figura 9 - Dual LS7366R Quadrature Encoder Buffer.....	13
Figura 10 – Placa de desenvolvimento Arduino Mega.....	15
Figura 11 - Diagrama de caso de uso - visão geral do sistema.	17
Figura 12 – Diagrama de caso de uso para o firmware do Arduino.	18
Figura 13 - Diagrama de classes utilizados no firmware.....	19
Figura 14 – Exemplo de arquitetura de rede.....	20
Figura 15 – Quadro geral de mensagem do MODBBUS.	21
Figura 16 –Quadro de mensagem do MODBUS Serial ASCII.	21
Figura 17 - Diagrama de caso de uso para o protocolo MODBUS.....	22
Figura 18 – MATLAB® GUI (<i>Graphical User Interfaces</i>).....	24
Figura 19 – Diagrama de caso de uso - visão geral do sistema.....	25
Figura 20 - Diagrama de caso de uso para a ação configuração.....	26
Figura 21 - Diagrama de caso de uso para a ação ensaios.....	26
Figura 22 – Diagrama de caso de uso para a ação operação.....	27
Figura 23 –Diagrama de caso de uso para a ação exibir gráficos.	27
Figura 24 – Interface do programa Plataforma XY.....	28
Figura 25 – <i>Panel</i> Configuração.....	30
Figura 26 – <i>Panel</i> Ensaios.	31
Figura 27 – <i>Panel</i> Operação.....	31
Figura 28 - <i>Panel</i> Gráficos.....	32
Figura 29- a) diagrama para o sistema eletromecânico para um eixo e b) diagrama de blocos representando o funcionamento do sistema.	34
Figura 30 - Diagrama de blocos para o controlador On/Off.	35
Figura 31 – a) PID em tempo contínuo e b) PID em tempo discreto.....	36
Figura 32 – Diagrama para motor CC.....	37
Figura 33 – Diagrama de bloco para a função de transferência do motor CC.....	38

Figura 34 – a) planta com controlador On/Off e b) planta com controlador PID discreto.....	39
Figura 35 - Ensaio em malha para ambos os eixos.....	41
Figura 36 – Comparativo entre os sinais estimados e medidos.....	42
Figura 37 – Ensaio em malha fechada utilizando o controlador On/Off.....	44
Figura 38 – Ensaio em malha fechada para os ganhos mostrados na Tabela 5.....	45
Figura 39 - Sinais de referência, ação de controle e saída para o ensaio em malha fechada utilizando controlador On/Off. a) Delta = 0,02 mm e b) Delta =0,2 mm.....	46
Figura 40 - Utilização do controlador On/Off para seguir uma onda quadrada como sinal de referência.....	46
Figura 41 - Sinais de referência, ação de controle e saída para o ensaio em malha fechada utilizando controlador PID. a) Degrau de 50 mm e b) Degrau de 250 mm.....	48
Figura 42 – Utilização do controlador PID para seguir uma onda quadrada como sinal de referência.....	48
Figura 43 – Circuito elétrico da planta.....	52

LISTA DE TABELAS E QUADROS

TABELAS

Tabela 1 – Especificações gerais do motor.	9
Tabela 2 - Principais características elétricas do driver VNH3SP30.	11
Tabela 3 – Características do Dual LS7366R.	13
Tabela 4 – Endereço dos registradores mapeados pelo MODBUS.	23
Tabela 5 - Ganhos dos controladores PID para os eixos.	43

QUADROS

Quadro 1 - Pinos de ligação do encoder.	10
Quadro 2 – Principais características técnicas da placa Arduino Mega.	14

SUMÁRIO

AGRADECIMENTO	iv
RESUMO	v
LISTA DE FIGURAS	vi
LISTA DE TABELAS E QUADROS	viii
SUMÁRIO	ix
1 Introdução	1
1.1 O Plano de estágio.....	1
1.1.1 Principais atividades.....	1
1.1.2 Cronograma	2
2 Local do Estágio	3
3 Atividades Realizadas	5
3.1 Visão geral sobre a plataforma didática.....	5
3.2 Estrutura física e circuitos eletrônicos	8
3.2.1 Estrutura física	8
3.2.2 Circuito de alimentação.....	8
3.2.3 Motor de corrente contínua com encoder	9
3.2.4 Driver para motor CC	10
3.2.5 Expansor para contador de pulsos e chaves de final de curso	12
3.2.6 Arduino Mega.....	14
3.3 Desenvolvimento do programa Plataforma XY	15
3.3.1 Firmware	17
3.3.2 Comunicação entre o computador e a planta didática	20
3.3.3 Desenvolvimento da interface	24
4 Plataforma XY	29
4.1 Descrição das funções da interface.....	29
4.1.1 Configuração.....	29
4.1.2 Ensaio	30
4.1.3 Operação	31
4.1.4 Gráficos.....	32
4.2 Identificação e controle da planta didática.....	33
4.2.1 Modelagem matemática	33
4.2.2 Ensaio em malha aberta	39
4.2.3 Identificação do processo.....	41

4.2.4	Sintonia dos controladores	42
4.2.5	Ensaio em malha fechada	43
4.3	Resultados	45
4.4	Melhorias sugeridas	49
5	Conclusão	50
6	Bibliografia	51
	APÊNDICE	52

1 Introdução

O objetivo deste relatório é apresentar as atividades desenvolvidas e os resultados alcançados pelo estagiário Christian Charles Dias, durante o estágio supervisionado realizado no Laboratório de Instrumentação Eletrônica e Controle (LIEC), situado na Universidade Federal de Campina Grande (UFCG). O período de vigência do estágio supervisionado foi de 11/01/2016 até 06/05/2016 totalizando 180h (6 créditos), sendo uma carga horária de 12 horas semanais, sob a orientação do professor George Acioli Júnior e supervisão do professor Péricles Rezende Barros.

1.1 O Plano de estágio

A elaboração do plano de estágio foi concebida de forma a possibilitar a concepção de uma planta didática para identificação e controle de um posicionador de dois eixos, requerendo do estagiário a aplicação dos conhecimentos adquiridos nas diversas disciplinas integralizadas durante a graduação, tendo uma maior aplicação os temas abordados nas disciplinas de Introdução à Programação, Técnicas de Programação, Eletrônica, Arquitetura de Sistemas Digitais, Controle Analógico, Controle Digital e Automação Industrial.

1.1.1 Principais atividades

1. Instrumentação da planta;
2. Desenvolvimento das funções para acionamento dos motores e leitura dos sensores via Arduino Mega;
3. Desenvolvimento do protocolo de comunicação MODBUS ASCII para Arduino;
4. Desenvolvimento do protocolo MODBUS ASCII para software MATLAB®;
5. Desenvolvimento das funções no MATLAB® para controle dos motores e leitura dos sensores;
6. Elaboração da interface gráfica no MATLAB®;
7. Identificação da planta utilizando o software MATLAB®;
8. Implementação dos controladores P, PI e PID;
9. Ensaios com o sistema em malha fechada;

1.1.2 Cronograma

Atividades	Janeiro			Fevereiro			Março			Abril			Maio		
1			■	■	■										
2				■	■										
3						■	■								
4							■	■							
5									■						
6										■	■	■	■	■	
7												■			
8													■	■	
9														■	

2 Local do Estágio

Localizado na Universidade Federal de Campina Grande (UFCG) no campus de Campina Grande, o Laboratório de Instrumentação Eletrônica e Controle (LIEC) é um laboratório pertence ao Departamento de Engenharia Elétrica (DEE). Integrado por professores doutores, alunos de pós-graduação e de graduação, esse laboratório tem como principal objetivo desenvolver atividades e projetos ligados as áreas de automação, controle e instrumentação.

Figura 1 – Fachada do LIEC.



Fonte: <http://lic.ufcg.edu.br/>, acessado em 11/05/2016

Com uma área de aproximadamente 600 m², o LIEC (Figura 1) conta com oito laboratórios de desenvolvimento, duas salas de apoio técnico, sala para apresentação de trabalhos, salas para pós-graduação e professores. Dentre as principais atividades desenvolvidas no laboratório, pode-se destacar as seguintes (LIEC, 2015):

- Laboratório de Aplicações Wireless - desenvolvimento de soluções baseadas em dispositivos móveis para ambientes industriais;
- Laboratório de Automação Industrial - sintonia de controladores PID industriais (mono e multivariável), automação industrial, instrumentação industrial, IHM industrial, avaliação de confiabilidade em malhas de controle;
- Laboratório de Controle e Otimização - projeto e sintonia de PID, modelagem e simulação de processos, sistemas supervisórios;
- Laboratório de Instrumentação Eletrônica - projeto e sintonia de PID;

- Laboratório de Redes Industriais - estudos de técnicas e tecnologias para a comunicação entre dispositivos industriais;
- Laboratório de RFID - desenvolvimento de aplicações baseadas em tecnologia RFID para ambientes industriais;
- Laboratório de Ultrassom - desenvolvimento de sensor de incrustação, desenvolvimento de técnicas de medição de incrustação.

No LIEC, alunos de pós-graduação e de graduação encontram um ambiente propício ao aperfeiçoamento dos conhecimentos teóricos e das habilidades práticas, por intermédio das diversas pesquisas e outras atividades realizadas no mesmo.

3 Atividades Realizadas

Conforme mencionado anteriormente, as atividades do estágio foram determinadas de forma a proporcionar ao estagiário um ambiente capaz de solidificar os conhecimentos adquiridos no decorrer da graduação, oferecendo um ambiente de trabalho, no qual, esses conhecimentos deveriam ser utilizados na elaboração e uma solução para um problema real de engenharia.

O estagiário foi encarregado de realizar o desenvolvimento de uma planta didática para identificação e controle de um posicionador de dois eixos, possibilitando a realização de ensaios em malha aberta e em malha fechada, bem como, permitir que diferentes técnicas de identificação e controle sejam aplicadas.

Para o desenvolvimento da solução, pode-se relacionar as atividades realizadas em quatro grupos:

- Instalação elétrica e elaboração de circuitos eletrônicos para acionamento e controle da planta didática;
- Desenvolvimento do firmware para o sistema embarcado;
- Desenvolvimento do software e interface para auxiliar o usuário a utilização da planta didática;
- Realização de ensaios, identificação e controle da planta didática.

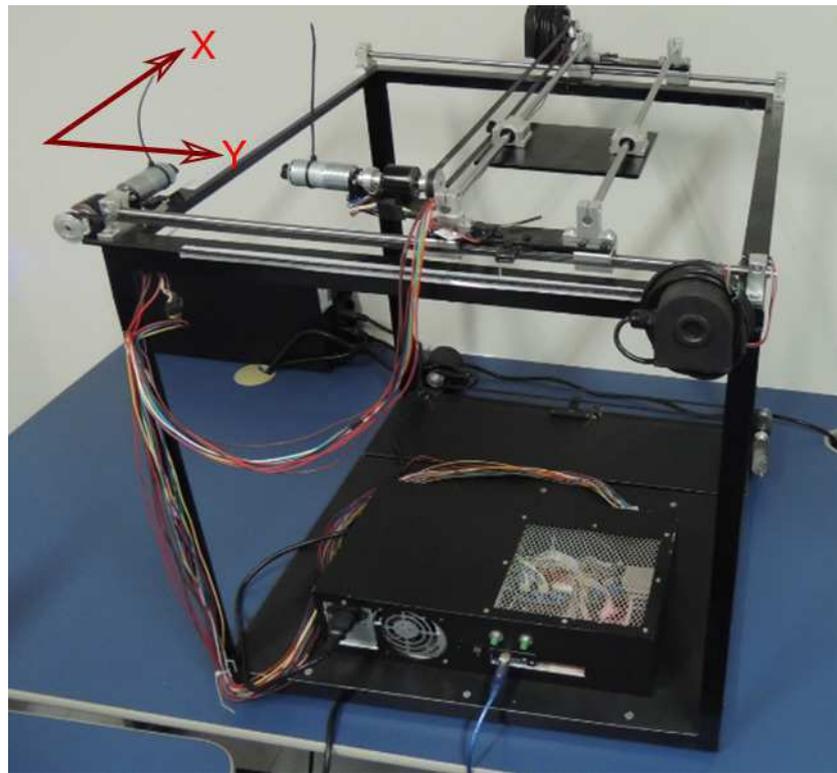
No decorrer desse relatório, esses grupos são descritos e explicados de forma detalhada.

3.1 Visão geral sobre a plataforma didática

A planta didática é um ambiente composto por um sistema de dois eixos cartesianos, no qual tem-se o deslocamento linear realizado por motores de corrente contínua controlados por um microcontrolador. Na Figura 2, é exibida a imagem da plataforma didática.

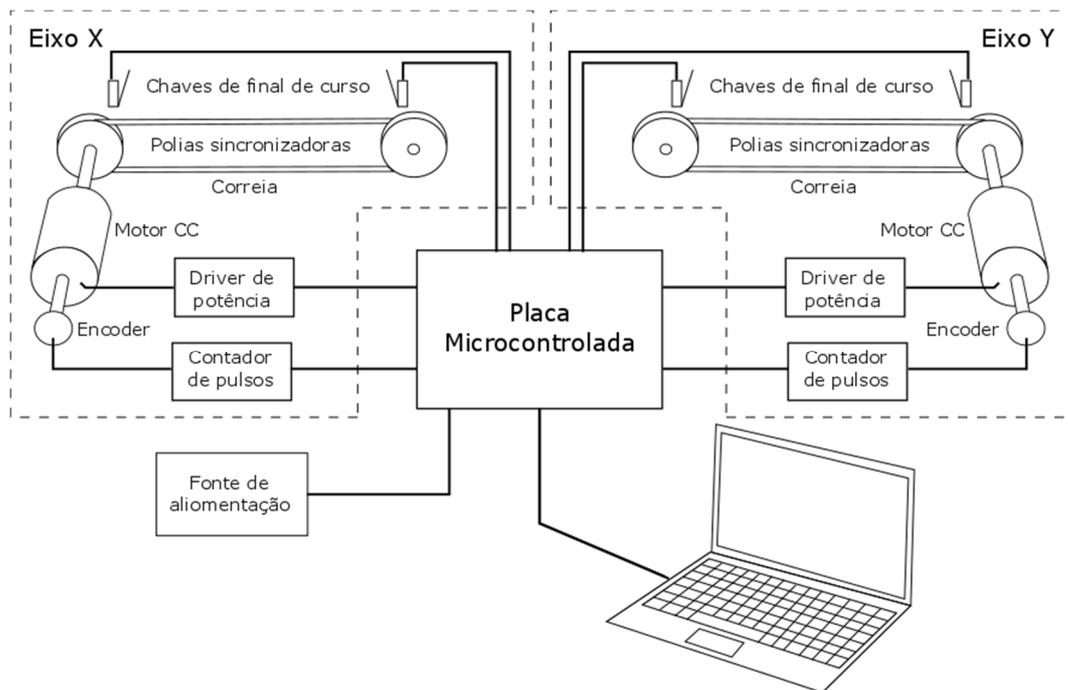
Observa-se na Figura 3, que ambos os eixos utilizam basicamente o mesmo conjunto de elementos eletromecânicos e sensores, sendo suas diferenças resguardadas por questões construtivas. A planta conta também com uma placa microcontrolada, que está conectada aos sistemas eletromecânicos, sensores e ao computador.

Figura 2 - Planta didática para identificação e controle de um posicionador de dois eixos.



Fonte: Próprio autor.

Figura 3 – Diagrama com os principais componentes da planta didática.



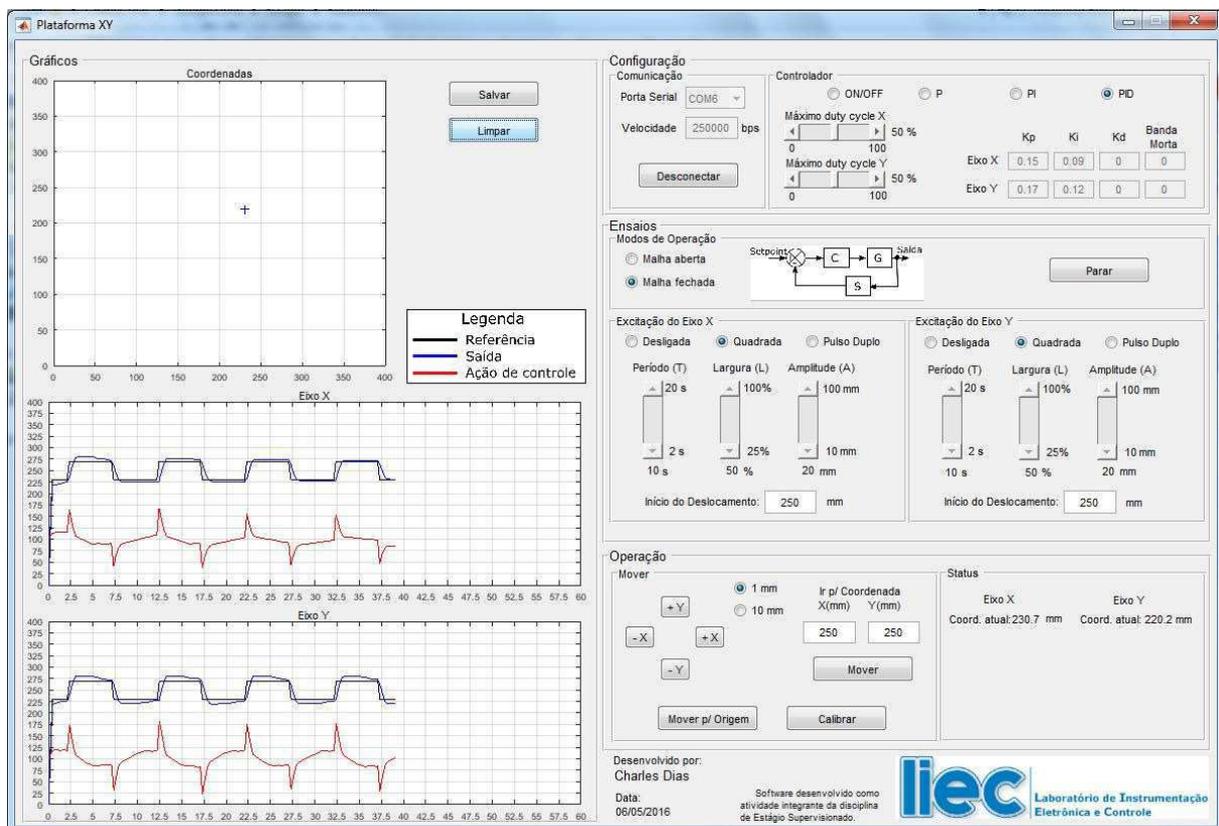
Fonte: Próprio autor.

Complementando a planta, é disponibilizado um software que realiza a integração entre o sistema eletromecânico microcontrolado e o usuário, oferecendo uma série de funcionalidades, tais como:

- Realizar procedimentos de configuração, como seleção da porta de comunicação, velocidade de comunicação, calibração do deslocamento, estados dos sensores, e outros.
- Efetuar ensaios em malha aberta e em malha fechada;
- Visualização dos dados em gráficos de posição versus tempo, e X versus Y;
- Selecionar o tipo de controlador, sem controlador, ON/OFF, P, PI ou PID, utilizado no ensaio de malha fechada;
- E ainda, salvar os dados referentes ao deslocamento dos eixos (coordenadas) e da ação de controle sobre os motores;

Na Figura 4 tem-se uma visão da interface, do software que foi desenvolvido, para realizar o controle da planta didática. Informações detalhadas sobre a elaboração e uso dos recursos disponíveis na interface, serão mostradas no decorrer do relatório.

Figura 4 – Interface da planta didática.



Fonte: Próprio autor.

3.2 Estrutura física e circuitos eletrônicos

3.2.1 Estrutura física

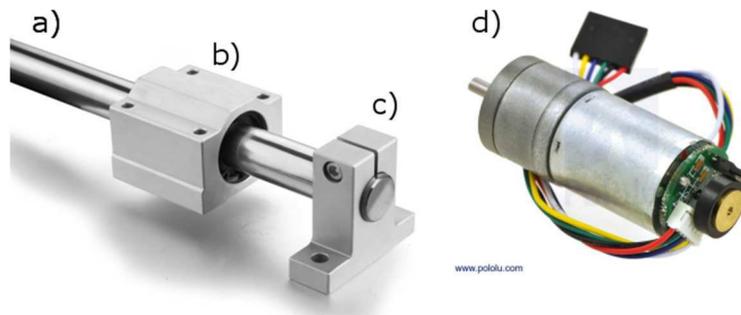
O projeto e montagem da estrutura física da planta didática não faz parte do escopo desse relatório, pois essas atividades foram realizadas por uma empresa contratada para tal. Assim, são descritas apenas as informações gerais sobre a concepção da estrutura física.

A estrutura é fabricada em aço galvanizado e cantoneiras de ferro em formato 'L', soldas de forma a definir um cubo nas dimensões de 55x55x55 cm, aproximadamente, conforme apresentada na Figura 2. Essa estrutura conta com dois sistemas de deslocamento, denominado aqui de eixos X e Y (plano cartesiano), compostas pelas seguintes partes:

- a) eixos de inox com 8 mm de diâmetro;
- b) rolamentos lineares ou *pillow block* linear;
- c) suporte para eixo ou linear *bearing rail*;
- d) motor de corrente contínua (motor CC) com encoder acoplado ao eixo.

Já na Figura 5, vê-se em detalhe as principais partes que constituem os eixos X e Y.

Figura 5 - Partes que compõem o eixo de deslocamento.



Fonte: Imagem adaptada pelo autor.

3.2.2 Circuito de alimentação

A alimentação da planta é provida por uma fonte chaveada tipo PC, no qual fornece tensão de +12 Vcc para alimentar a placa Arduino Mega, aplicada a sua entrada de alimentação principal, e para a fonte linear que alimenta o circuito de acionamento dos motores.

Sendo a tensão nominal dos motores CC de 6 Vcc, faz-se necessário incluir um circuito regulador de +12 Vcc para +6,6 Vcc, compatibilizando para o valor próximo da tensão nominal.

3.2.3 Motor de corrente contínua com encoder

Para realizar o deslocamento do conjunto mecânico, foi decidido por utilizar motores CC com encoders, por efeito hall, acoplados diretamente ao eixo do motor. Foi selecionado o modelo 75:1 Metal Gearmotor 25Dx54L mm LP 6V with 48 CPR Encoder, apresentado na Figura 6, comercializado pelo fornecedor Pololu Robotics & Electronics.

Figura 6 - 75:1 Metal Gearmotor 25Dx54L mm LP 6V with 48 CPR Encoder.



Fonte: <https://www.pololu.com/product/2286>.

Esse motor CC consiste em um modelo de baixo consumo com tensão nominal de 6 Vcc e uma caixa de redução acoplada, sendo sua relação de 78,83:1. O motor também conta com um encoder com resolução de 48 pulsos por volta integrado ao eixo do motor, que fornece 3591,84 pulsos por volta do eixo de saída da caixa de redução. As especificações gerais do motor são apresentadas na Tabela 1.

Tabela 1 – Especificações gerais do motor.

Relação de redução	74,83:1
Velocidade à vazio	78 rpm
Corrente à vazio	250 mA
Corrente com rotor bloqueado (6V)	2,4 A
Torque com rotor bloqueado (6V)	95 oz·in

Fonte: Adaptado do <https://www.pololu.com/product/2286>

O encoder (sensor hall) exige uma tensão de entrada entre 3,5 e 20 Vcc e consome uma corrente máxima de 10 mA. As saídas A e B são ondas quadradas de

0 V para Vcc com aproximadamente 90° de defasagem. Medindo a frequência das transições tem-se a velocidade do motor, e medindo a ordem das transições tem-se a direção. No Quadro 1, são descritas as respectivas funções dos pinos do encoder.

Quadro 1 - Pinos de ligação do encoder.

Cor	Função
Vermelho	Alimentação do motor
Preto	Alimentação do motor
Verde	GND do encoder
Azul	Vcc do encoder (3,5 – 20 V)
Amarelo	Saída A do encoder
Branco	Saída B do encoder

Fonte: Adaptado do <https://www.pololu.com/product/2286>

3.2.4 Driver para motor CC

Para acionamento dos motores CC, optou-se por trabalhar com módulo Dual VNH2SP30 Motor Driver Carrier MD03A do fornecedor Pololu Robotics & Electronics, apresentado na Figura 7.a. Esse módulo possibilita realizar o controle de dois motores de alta potência, corrente de até 14 A, mediante a utilização de uma ponte H, propiciando o acionamento e a reversão no sentido de rotação. Além disso, o driver de potência utilizado nesse módulo (VNH3SP30) possibilita o monitoramento da corrente que está sendo acionada, fornecendo uma saída de tensão proporcional ao valor da corrente, e também conta com proteção de sobretensão.

Realizar o acionamento de motores por intermédio desse módulo torna-se uma tarefa bastante simples. Estando os pinos VCC e VIN (Figura 7.b) devidamente conectados as tensões de +5 Vcc e +6,6 Vcc, respectivamente, pode-se realizar o acionamento do motor mediante os seguintes passos:

1. selecionar o sentido de rotação, sentido de circulação da corrente na ponte H, atuando no pinos INA e INB (Figura 7.b);
2. entrar com o sinal PWM, que será responsável por determinar a velocidade de rotação do motor, no pino PWM (Figura 7.b);
3. o estado de funcionamento do módulo pode ser monitorado mediante o acompanhamento do nível de tensão nos pinos ENA/DIAGA e ENB/DIAGB (Figura 7.b), essa funcionalidade não foi utilizada no projeto;

4. o valor aproximado da corrente, que está sendo acionada, por ser medida mediante a leitura da tensão presente no pino CS (Figura 7.b) por um fator de conversão de 0,13 V/A.

As principais características elétricas para o driver de potência VNH3SP30, estão apresentadas na Tabela 2.

Tabela 2 - Principais características elétricas do driver VNH3SP30.

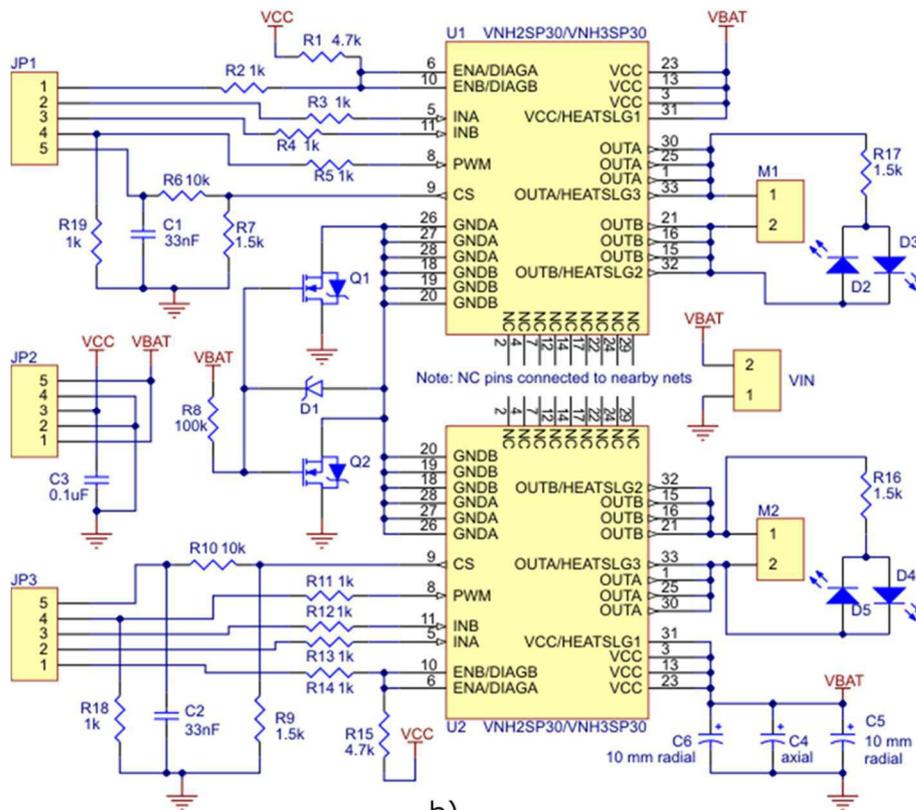
Tensão de alimentação (Vcc)	5,5 - 16 V
Máxima corrente de atuação (temporária)	30 A
Resistência do <u>MOSFET</u>	19 mΩ
Máxima frequência do PWM	20 kHz
Sensor de corrente	Aproximadamente 0,13 V/A
Desligamento por sobretensão	Mínimo de 16 V (típico de 19 V)
Corrente acionada sem limite de tempo	14 A

Fonte: adaptado do <https://www.pololu.com/product/708>

Figura 7 - Dual VN2SP30 Motor Driver Carrier MD03A, a) Vista do módulo montado e b) Circuito eletrônico do módulo.



a)



b)

Fonte: <https://www.pololu.com/product/708>.

3.2.5 Expansor para contador de pulsos e chaves de final de curso

Foi desenvolvido uma placa de expansão (*shield*) para o Arduino Mega, permitindo o acoplamento do módulo Dual LS7366R Quadrature Encoder Buffer e conexão das chaves de final de curso. O layout da placa de expansão é apresentado na Figura 8.

O módulo Dual LS7366R Quadrature Encoder Buffer (Figura 9) é composto por dois contadores de 32 bits, projetado para interagir diretamente a uma saída do

3.2.6 Arduino Mega

Devido a popularização das placas de desenvolvimento baseadas em Arduino, foi decidido que o sistema eletromecânico deveria ser controlado por uma dessas placas baseadas em Arduino, possibilitando que futuras alterações no firmware fossem realizadas por outros alunos, sem demandar elevado tempo de aprendizagem.

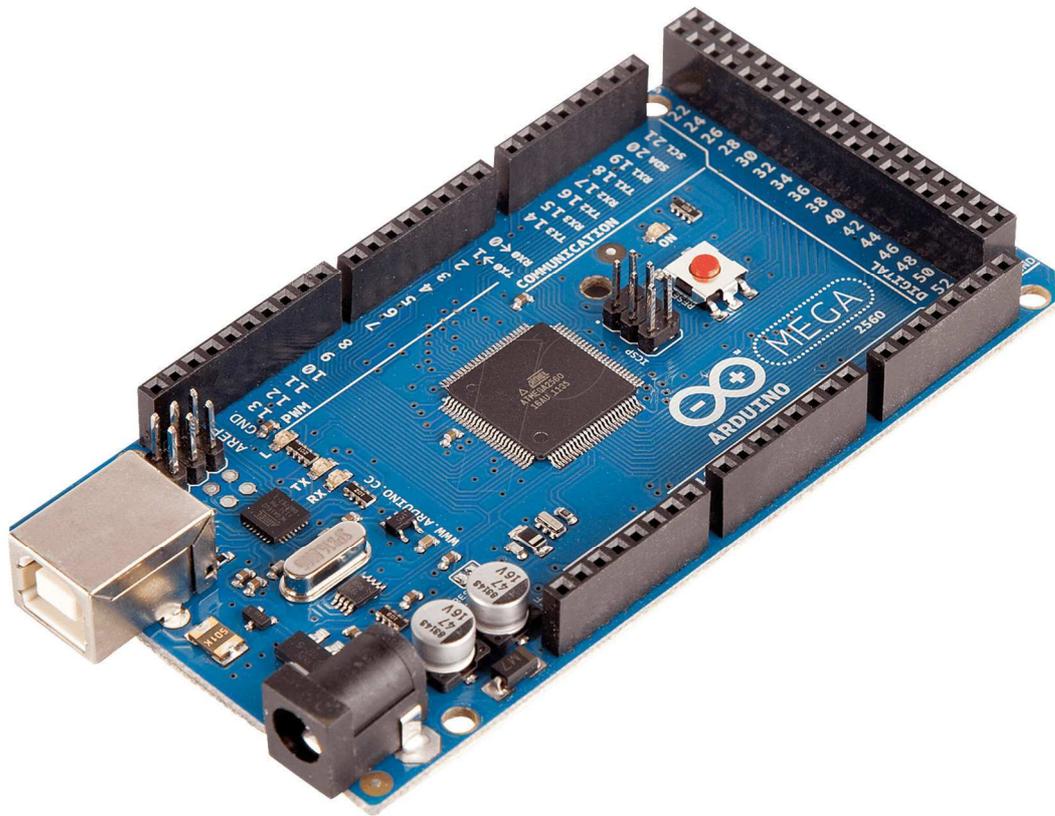
Dessa forma, foi selecionado o modelo Arduino Mega, apresentado na Figura 10, devido ao seu elevado número de recursos de hardware, quando comparado com outras placas da plataforma Arduino. No Quadro 2, tem-se as principais características técnicas para a placa utilizada.

Quadro 2 – Principais características técnicas da placa Arduino Mega.

Microcontrolador	ATmega2560
Tensão de operação	+5 Vcc
Tensão de entrada (recomendado)	(7 – 12 V)
Tensão de entrada (limites)	(6 – 20 V)
Pinos digitais entrada/saída	54
Entradas analógicas (10 bits de resolução)	16
Corrente por pino	20 mA
Memória Flash	256 kB
Memória SRAM	8 kB
Memória EEPROM	4 kB
Oscilador do sistema	10 MHz
Comprimento	101,52 mm
Largura	53,3 mm

Fonte: Adaptado de <https://www.arduino.cc/en/Main/arduinoBoardMega2560>.

Figura 10 – Placa de desenvolvimento Arduino Mega.



Fonte: <https://arduino-info.wikispaces.com/MegaQuickRef>.

Observação: O esquema elétrico da planta encontra-se no apêndice.

3.3 Desenvolvimento do programa Plataforma XY

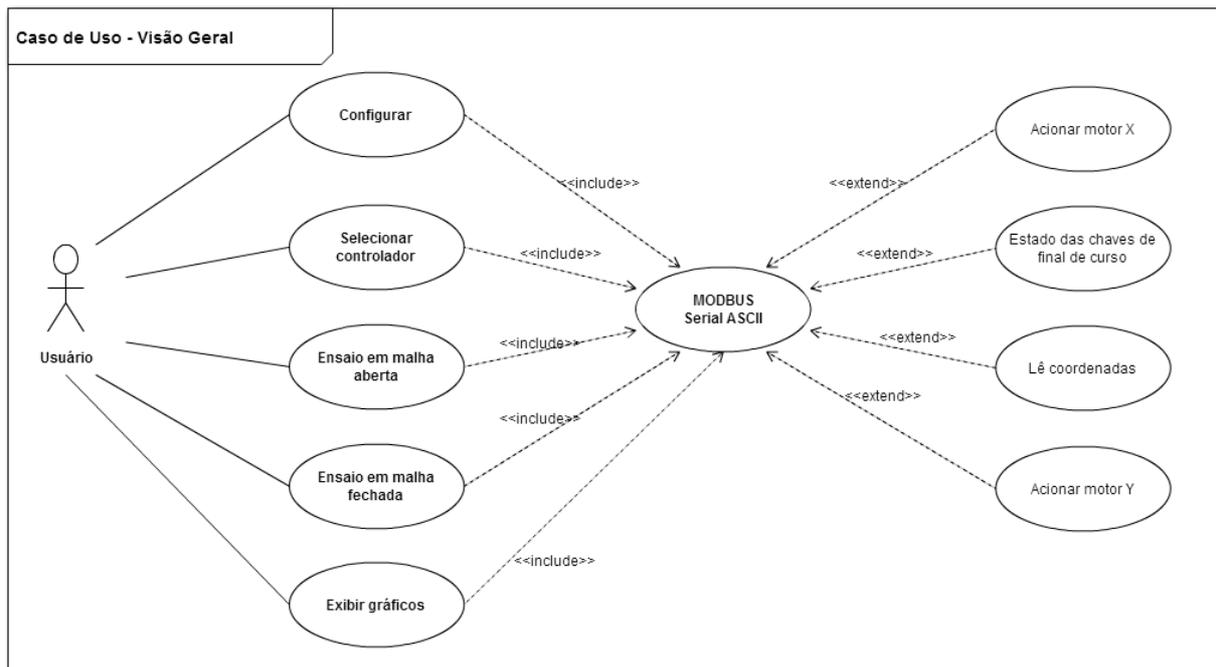
Foi desenvolvido o programa chamado de Plataforma XY, para possibilitar a integração entre a planta e o usuário, permitindo realizar ações sobre a planta e monitorar os efeitos dessas ações. Na Figura 11, tem-se o diagrama de caso de uso com a visão geral do funcionamento do programa.

No diagrama de uso, pode-se verificar a existência de diversas ações que podem ser realizadas pelo usuário, sendo elas detalhadas a seguir:

- Configurar – essa ação permite realizar a seleção da porta de comunicação serial e velocidade de comunicação, calibração do deslocamento e mover o conjunto eletromecânico para a origem;
- Selecionar controlador – permite selecionar o tipo de controlador que atuará sobre o deslocamento dos sistemas eletromecânicos. Os tipos de controladores disponibilizados são On/Off, P, PI ou PID, podendo ser escolhidos tanto para a malha aberta quanto para malha fechada.

- Ensaio em malha aberta – pode-se selecionar a forma de onda da excitação, onda quadrada ou pulso duplo, configurando o tempo de período, a amplitude do *duty cycle* e a largura do pulso, no caso da excitação ser um pulso duplo;
- Ensaio em malha fechada – possui opções bastantes semelhantes ao ensaio em malha aberta, só que neste caso, a configuração será para a onda que servira como sinal de referência;
- MODBUS Serial ASCII – realiza a comunicação entre o software que está funcionando no computador e o firmware que está gravado no Arduino Mega;
- Acionar motor X ou Acionar motor Y – essas duas ações são semelhantes, sendo a única diferença o eixo de atuação dela. Esta ação é responsável por realizar ações de ligar/desligar o motor, inverter sentido de rotação e desligar o motor, caso o limite de deslocamento seja atingido;
- Estado das chaves de final de curso – monitora o estado das chaves de final de curso, sinalizando para o microcontrolador quando o limite do deslocamento é atingido;
- Lê coordenadas – realiza funções relacionadas ao posicionamento da plataforma, como armazenar informações de posição atual e limpar o contador de pulso.

Figura 11 - Diagrama de caso de uso - visão geral do sistema.



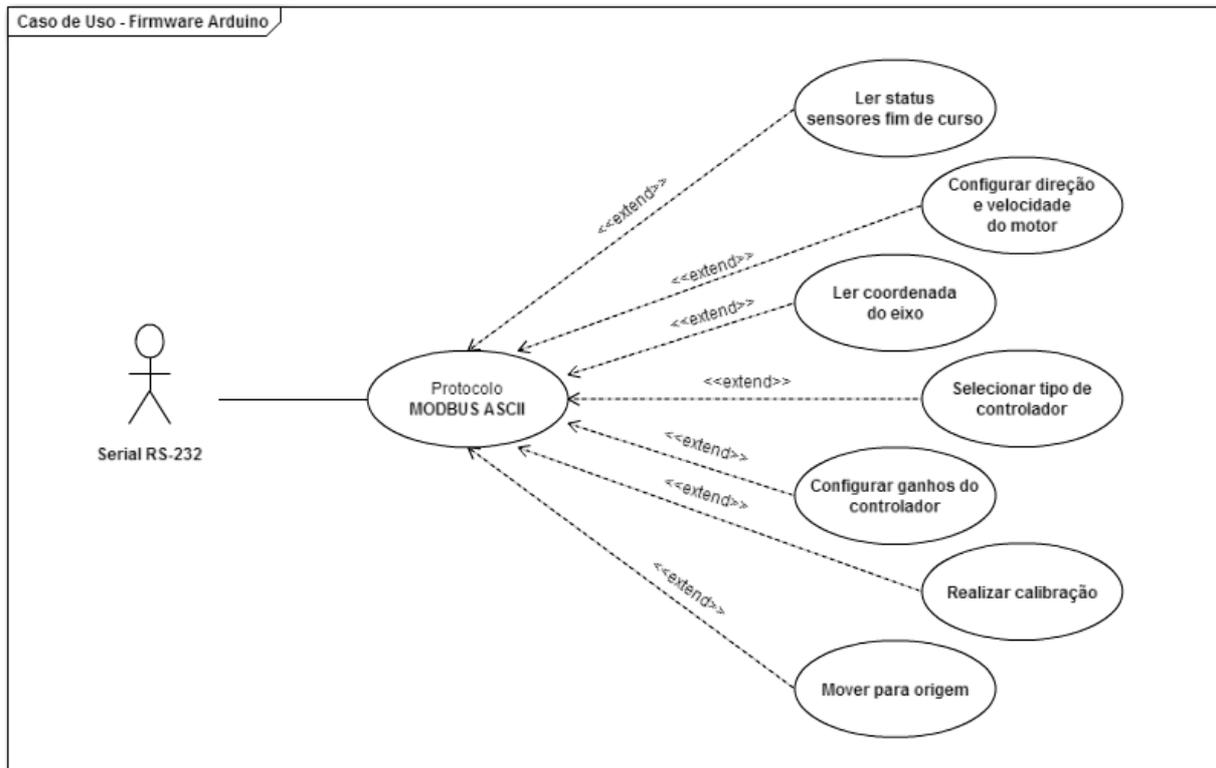
Fonte: Próprio autor.

Para atender os requisitos descritos anteriormente, o software foi desenvolvido em dois níveis: o primeiro nível ficou encarregado de realizar a interface com o hardware, oferecendo uma camada de abstração de hardware (HAL - Hardware Abstract Layer); o segundo nível ficou encarregado de realizar a interface com o usuário, permitindo que as ações, já descritas anteriormente, fossem realizadas sem que o usuário precise de maiores conhecimentos sobre como o sistema realmente funciona.

3.3.1 Firmware

O firmware implementado para o Arduino ficou encarregado de realizar a abstração do hardware, de forma que comandos recebidos via a serial RS-232, sob o protocolo de comunicação MODBUS Serial ASCII, pudesse realizar as ações apresentadas no diagrama de caso de uso da Figura 12.

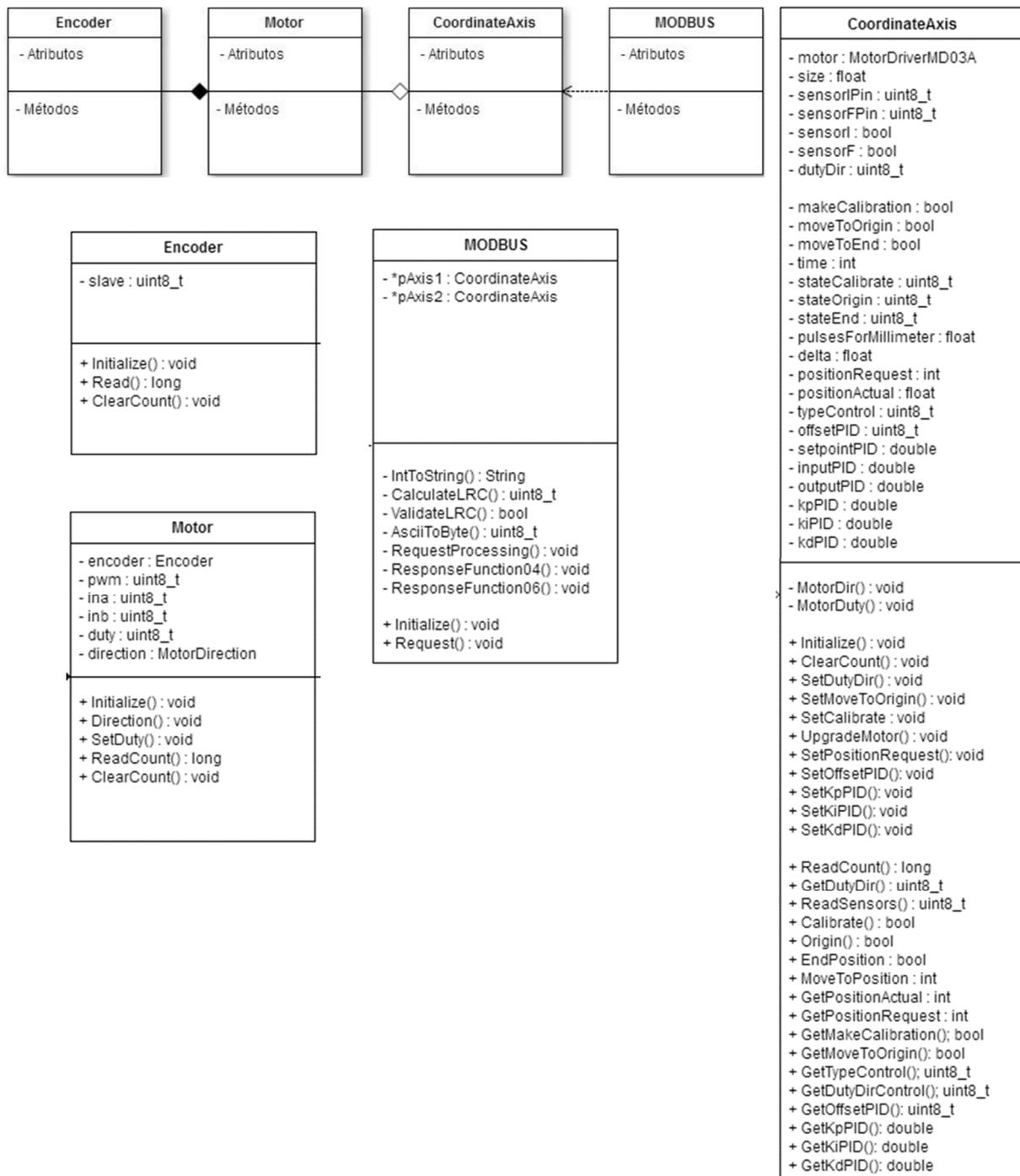
Figura 12 – Diagrama de caso de uso para o firmware do Arduino.



Fonte: Próprio autor.

Tomando como base os elementos eletromecânicos constituintes do sistema de deslocamento e analisando o diagrama de uso apresentado, foram criados quatro objetos para realizar abstração dos encoder, motor, eixo coordenado e protocolo MODBUS. Na Figura 13 são apresentadas as classes criadas, bem como, as relações entre elas.

Figura 13 - Diagrama de classes utilizados no firmware.



Fonte: Próprio autor.

Na classe *Encoder*, são disponibilizados métodos para inicializar o objeto, lê o valor do contador e limpar o contador. A classe *Motor*, que possui uma relação de composição com a classe *Encoder*, fornece métodos para inicializar o objeto motor, configurar a direção e a velocidade. Já na classe *MODBUS*, encontram-se métodos referentes ao protocolo de comunicação MODBUS Serial ASCII. E por fim, tem-se a

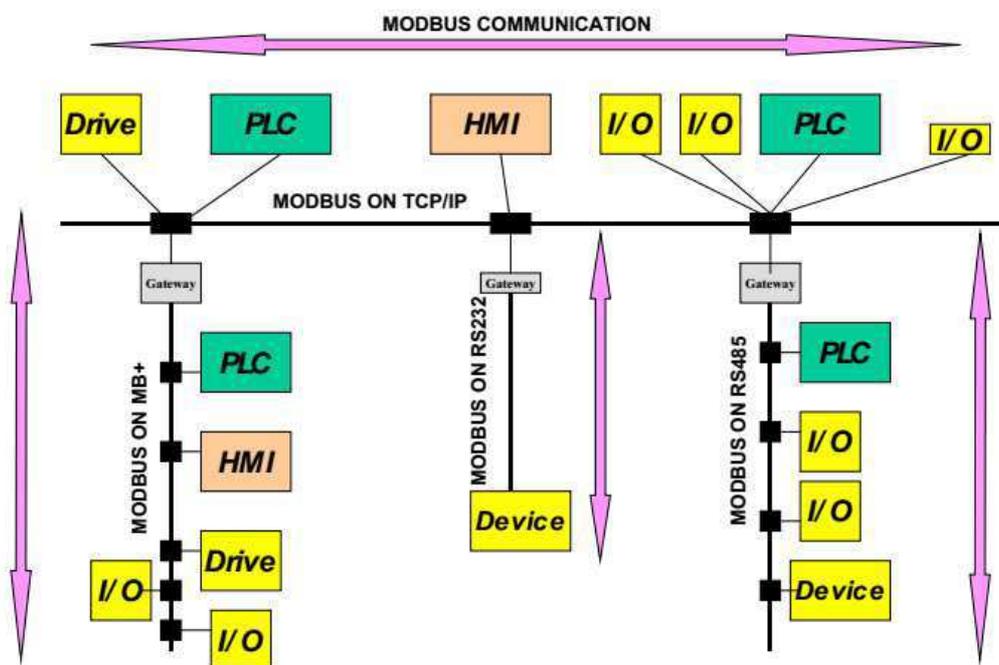
classe *CoordinateAxis*, que possui relação de agregação com a classe *Motor* e de dependência com a classe *MODBUS*, disponibilizando várias funções pertinentes ao funcionamento do objeto eixo.

3.3.2 Comunicação entre o computador e a planta didática

A comunicação entre o computador e a planta é realizada por intermédio da comunicação USB, que implementa uma serial virtual RS-232. Assim, a comunicação entre o computador e a planta didática ocorre sobre essa camada de abstração, funcionando como se estivessem conectadas fim a fim sob uma comunicação serial RS-232.

Sobre a camada física de comunicação, foi implementado o protocolo de comunicação MODBUS Serial. Sua escolha se deu principalmente por ser um protocolo utilizado em diversos equipamentos de aplicação industrial, tendo a licença *open source* para essa versão de MODBUS Serial; e a vasta existência de conteúdo abordando a implementação do mesmo. Na Figura 14, pode-se observar uma outra característica do protocolo que é a concepção de uma arquitetura de rede com diferentes tipos de camada física, na qual, mediante os gateways, permite que os dispositivos se comuniquem entre si.

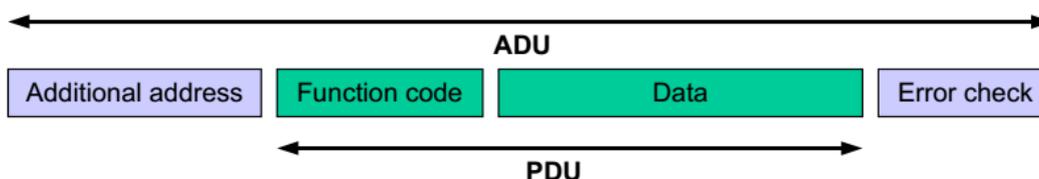
Figura 14 – Exemplo de arquitetura de rede.



Fonte: MODBUS APPLICATION PROTOCOL SPECIFICATION
V1.1b3.

O protocolo MODBUS define uma simples unidade de protocolo de dados (PDU) que é independente das camadas de comunicação subjacentes. Conforme pode ser observado na Figura 15, na qual se apresenta a estrutura geral do quadro de mensagem, o PDU é formado por duas partes, sendo a primeira referente ao código da função (*Function code*), e a segunda, referente ao campo de dados (*Data*). Adicionado ao PDU, verifica-se ainda mais dois campos, o de endereço (*Additional address*) e o de checagem de erros (*Error check*), respectivamente.

Figura 15 – Quadro geral de mensagem do MODBUS.



Fonte: MODBUS APPLICATION PROTOCOL SPECIFICATION
V1.1b3.

No protocolo MODBUS Serial é disponibilizado dois modos de utilização: o ASCII e o RTU. Foi definido que seria utilizado o modo ASCII, devido a sua facilidade de implementação, face ao modo RTU, uma vez que não se tem a forte dependência do tempo entre os quadros de mensagem, característica presente no modo RTU. Porém, essa escolha traz uma desvantagem, que é a pequena densidade de dados, pois a comunicação é realizada mediante o envio de caracteres ASCII. Na Figura 16, é apresentado a estrutura do quadro de mensagem do MODBUS Serial ASCII.

Figura 16 –Quadro de mensagem do MODBUS Serial ASCII.

Start	Address	Function	Data	LRC	End
1 char :	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CR,LF

Fonte: MODBUS over serial line specification and implementation guide V1.02.

Os campos presentes na estrutura do quadro de mensagem, correspondem as seguintes funções:

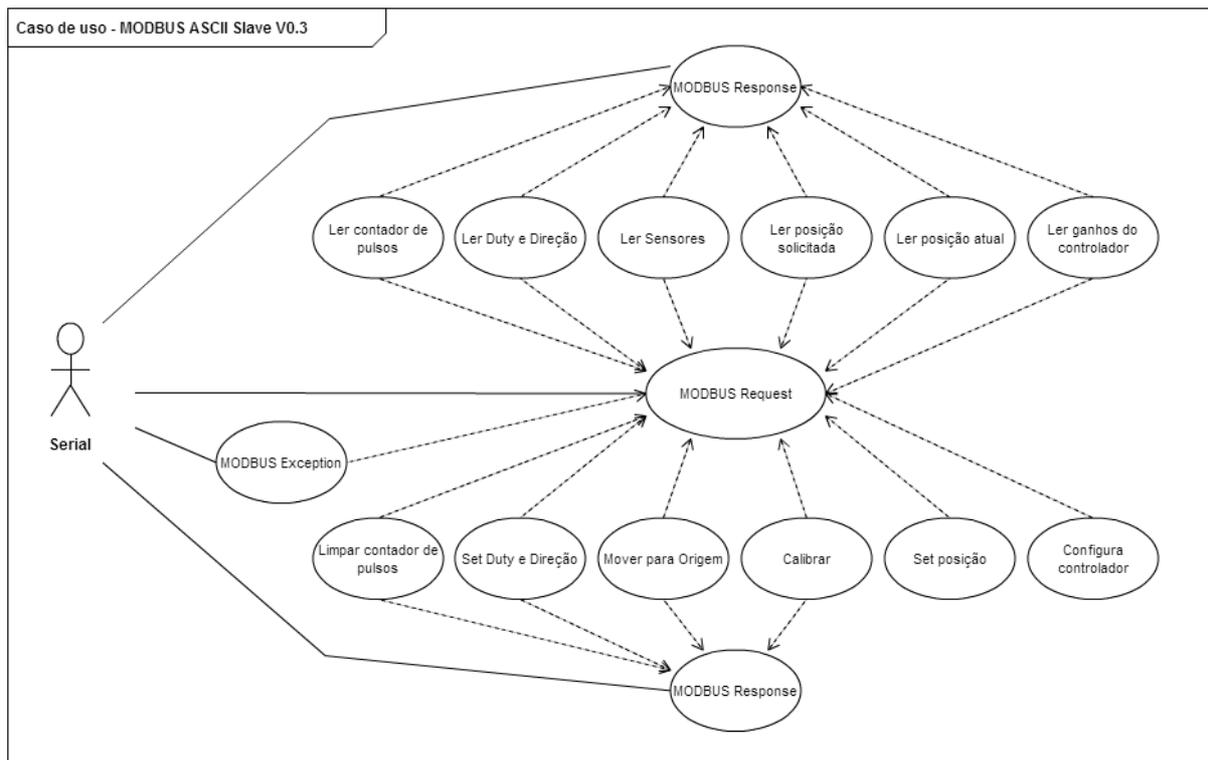
- *Start* – o caractere ':' (0x3A) sinaliza o início do quadro de mensagem;
- *Address* – referente ao endereço do dispositivo alvo;
- *Function* – função definida no protocolo. Nessa aplicação, apenas as funções 0x04 e 0x06 foram implementadas, leitura de registrador e escrita em único registrador, respectivamente;

- *Data* – esse campo consta efetivamente a informação trafegada;
- *LRC* – algoritmo de checagem de erro;
- *End* – campo que sinaliza o final do quadro de mensagem.

Com relação ao quadro de mensagem, é importante salientar que à exceção dos campos de *Start* e *End*, apenas os caracteres ‘0’, ‘1’, ... ‘9’, ‘A’, ‘B’, ... ‘F’ são considerados válidos nos campos da mensagem.

Seguindo as recomendações presentes nas documentações do protocolo, foi desenvolvida uma classe MODBUS, já apresentada na Figura 13, permitindo que ao realizar a comunicação com a planta, o software que funciona no computador, pudesse ter acesso as variáveis de interesse. Para orientar o desenvolvimento dessa classe, foi elaborado o diagrama de caso que está representado na Figura 17.

Figura 17 - Diagrama de caso de uso para o protocolo MODBUS.



Fonte: Próprio autor.

O protocolo MODBUS também foi implementado no programa Plataforma XY, possibilitando, mediante os registradores que realizam o mapeamento parte da memória do Arduino, ter acesso as variáveis correspondentes as configurações de funcionamento da planta didática.

Na Tabela 4, estão descritos os endereços dos registradores e as informações que eles portam. Salienta-se a existência de dois grupos de registradores: aqueles utilizados nas operações de leitura; e os que são utilizados para operações de escritas. Caso nas funções sejam utilizados endereços de registradores que não existem, o protocolo retornará um código de erro informando que o registrador não existe. E ainda, para uma operação de escrita, deve-se verificar a faixa de valores válido para cada registrador, pois ao tentar escrever um valor fora dessa faixa será retornado erro na operação.

Tabela 4 – Endereço dos registradores mapeados pelo MODBUS.

Endereço e função dos registradores acessíveis ao MODBUS			
	Endereço*	Nome*	Informação
Utilizados para Leitura	0xX001	GET_SENSORES_X	Estado das chaves de final de curso
	0xX002	READ_COUNT_X	Valor dos pulsos do contador
	0xX003	GET_DUTY_DIR_X	Valor do duty cycle e direção de rotação
	0xX006	GET_POSITION_REQUEST_X	Coordenada da posição solicitada
	0xX007	GET_POSITION_ACTUAL_X	Coordenada da posição atual
	0xX008	GET_TYPE_CONTROL_X	Tipo de controlador que está configurado
	0xX009	GET_DUTY_DIR_CONTROL_X	Valor do duty cycle utilizado para os controladores
	0xX00A	GET_OFFSET_PID_X	Configura valor de banda morta
	0xX00B	GET_KP_PID_X	Valor do ganho Kp
	0xX00C	GET_KI_PID_X	Valor do ganho Ki
	0xX00D	GET_KD_PID_X	Valor do ganho Kd
Utilizados para Escrita	0xX102	CLEAR_COUNT_X	Limpa valor do contador
	0xX103	SET_DUTY_DIR_X	Configura valor de duty cycle e direção de rotação
	0xX104	SET_CALIBRATE_X	Ativa rotina de calibração
	0xX105	SET_ORIGIN_X	Ativa rotina para mover para origem
	0xX106	SET_POSITION_REQUEST_X	Configura coordenada
	0xX108	SET_TYPE_CONTROL_X	Configura o tipo de controlador
	0xX109	SET_DUTY_DIR_CONTROL_X	Configura o valor do duty cycle utilizado no controlador
	0xX10A	SET_OFFSET_PID_X	Configura o valor de banda morta
	0xX10B	SET_KP_PID_X	Configura o ganho kp
	0xX10C	SET_KI_PID_X	Configura o ganho ki
	0xX10D	SET_KD_PID_X	Configura o ganho kd

Fonte: Próprio autor.

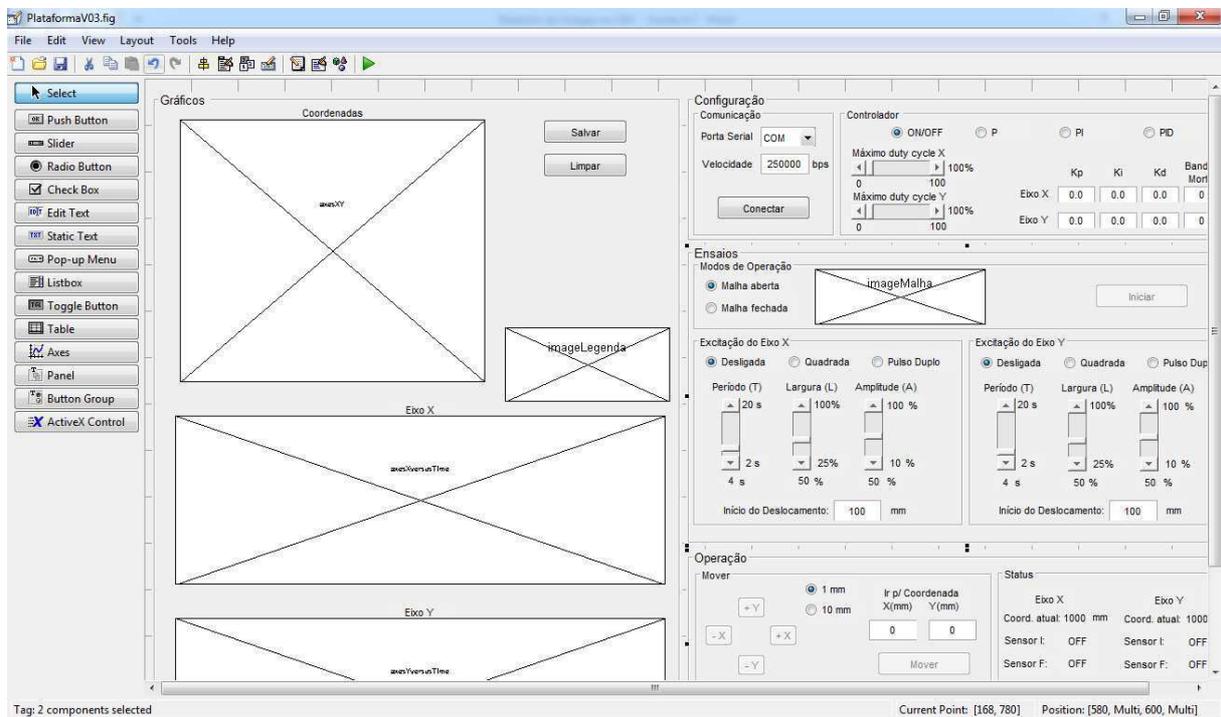
* : O valor 'X' presente nos campos endereço e nome, deve ser substituído pelo valor '1' quando for relativo ao eixo X e valor '2' quando for o eixo Y.

Observação: Maiores informações sobre a estrutura de funcionamento do protocolo MODBUS, podem ser encontradas nos documentos MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b e MODBUS Over Serial Line Specification and Implementation Guide V1.02.

3.3.3 Desenvolvimento da interface

Para desenvolver a interface do programa Plataforma XY foi utilizado o ambiente computacional MATLAB®, mais especificamente o MATLAB® GUI (*Graphical User Interfaces*). A GUI, podendo ser chamado também de UI (*User Interfaces*) fornece ferramentas para a criação de interfaces de usuário para aplicativos personalizados, gerando automaticamente o código MATLAB para construir a interface do usuário, que pode ser modificada para programar o comportamento de seu aplicativo. A GUI geralmente contém controles, como menus, barras de ferramentas, botões e controles deslizantes, exibidos na Figura 18.

Figura 18 – MATLAB® GUI (*Graphical User Interfaces*).

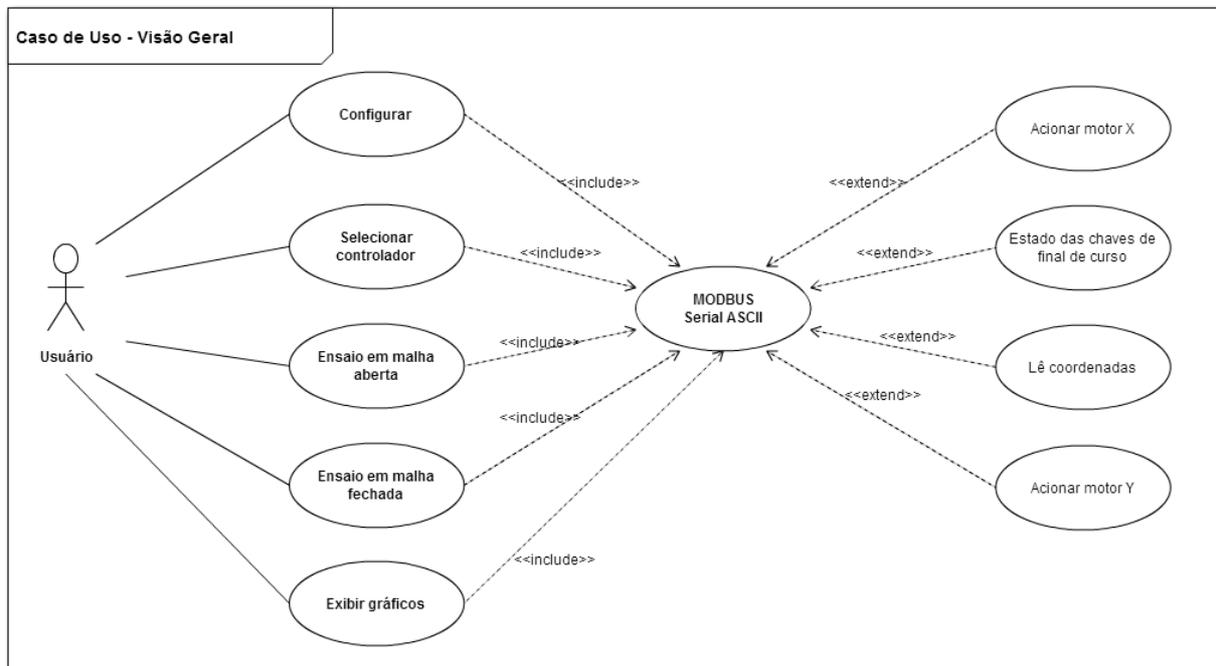


Fonte: Próprio autor.

Conforme descrito anteriormente, o programa realiza a interface entre o usuário e a planta didática, possibilitando que o mesmo realize o conjunto de atividades já

descritas. Na Figura 19 é reexibido o diagrama de caso de uso descrevendo o funcionamento geral do sistema.

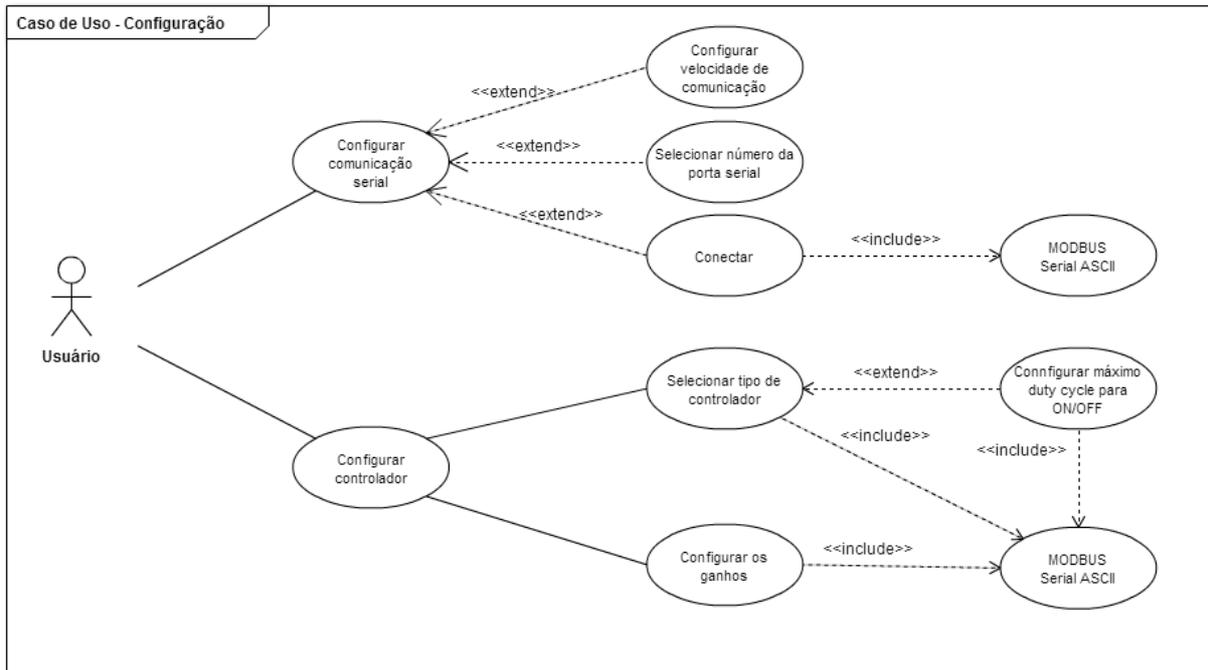
Figura 19 – Diagrama de caso de uso - visão geral do sistema.



Fonte: Próprio autor.

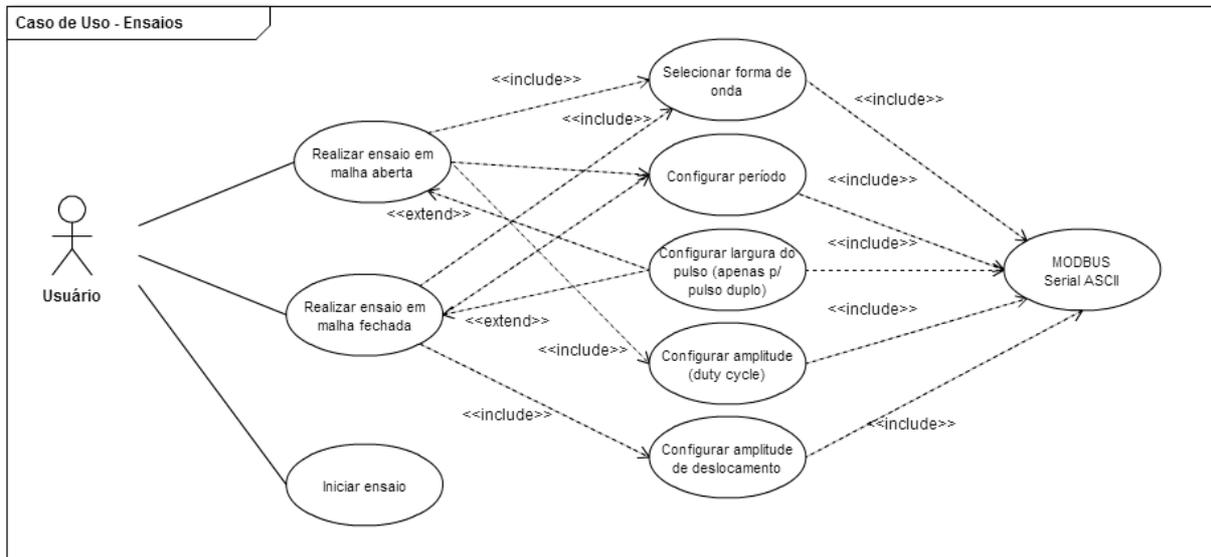
Para detalhar melhor as atividades exibidas no diagrama de caso da Figura 19, foram elaborados outros diagramas de caso de uso, descrevendo com maior detalhe as ações que deveriam ser realizadas pelo programa. Dessa forma, foram elaborados os diagramas apresentados nas Figura 20, 22, 23 e 24.

Figura 20 - Diagrama de caso de uso para a ação configuração.



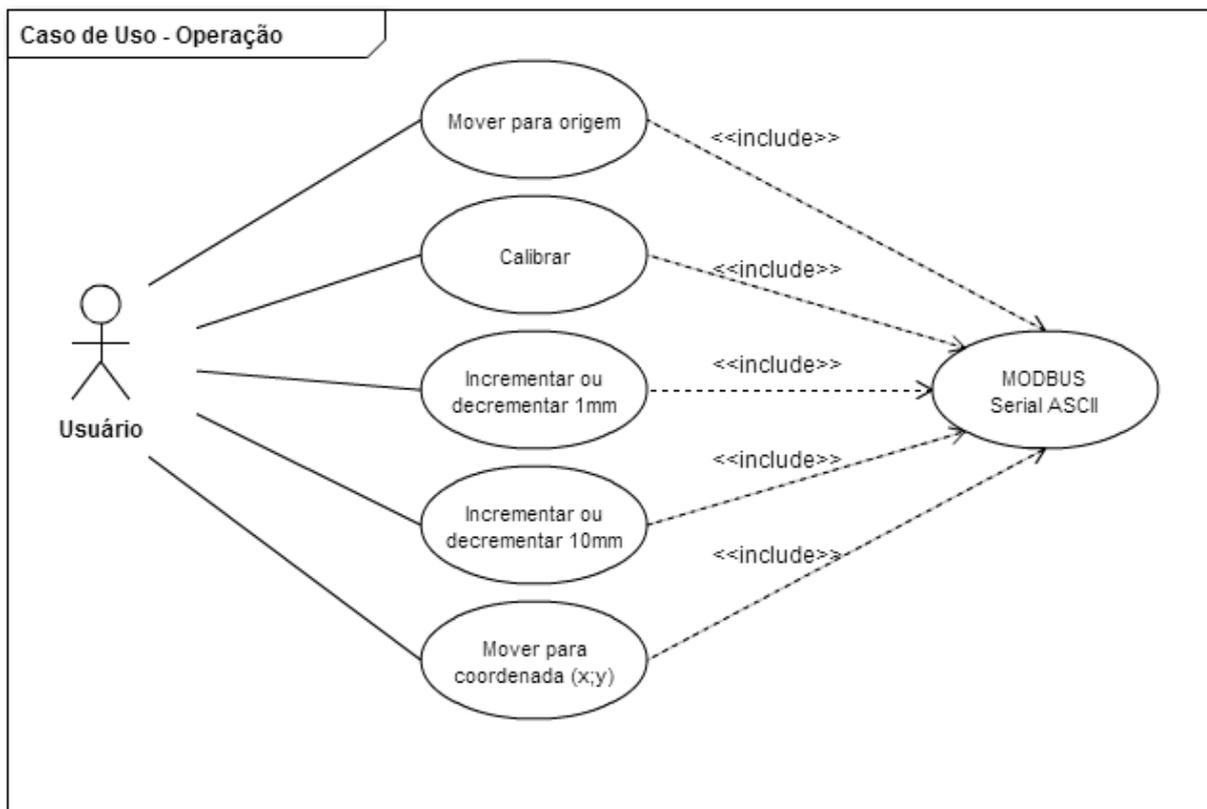
Fonte: Próprio autor.

Figura 21 - Diagrama de caso de uso para a ação ensaios.



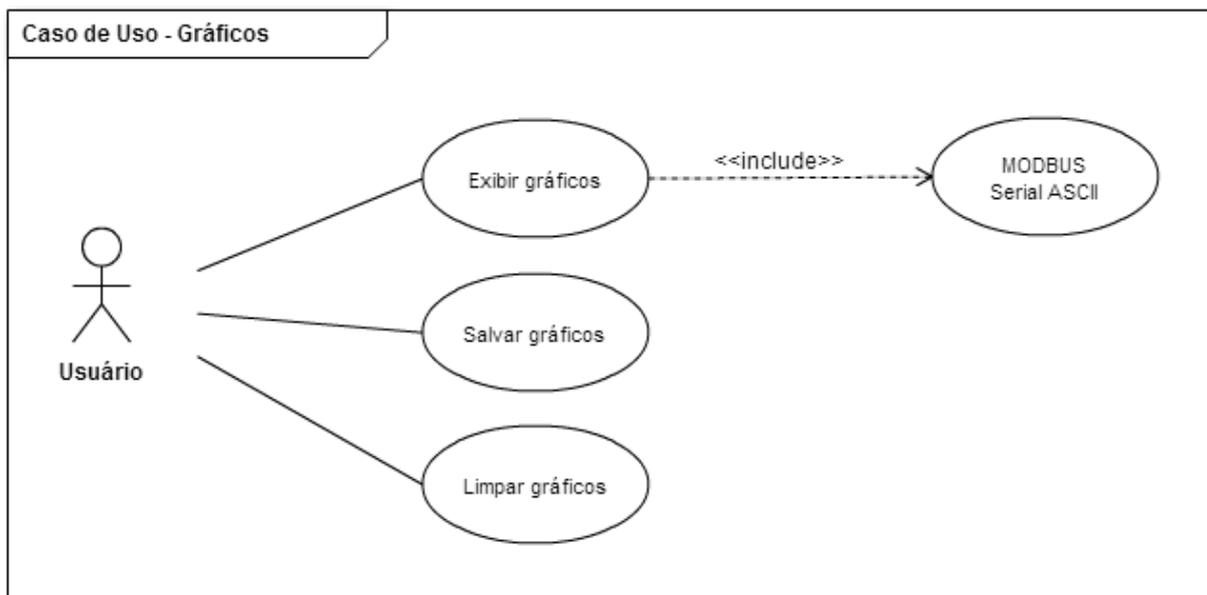
Fonte: Próprio autor.

Figura 22 – Diagrama de caso de uso para a ação operação.



Fonte: Próprio autor.

Figura 23 –Diagrama de caso de uso para a ação exibir gráficos.

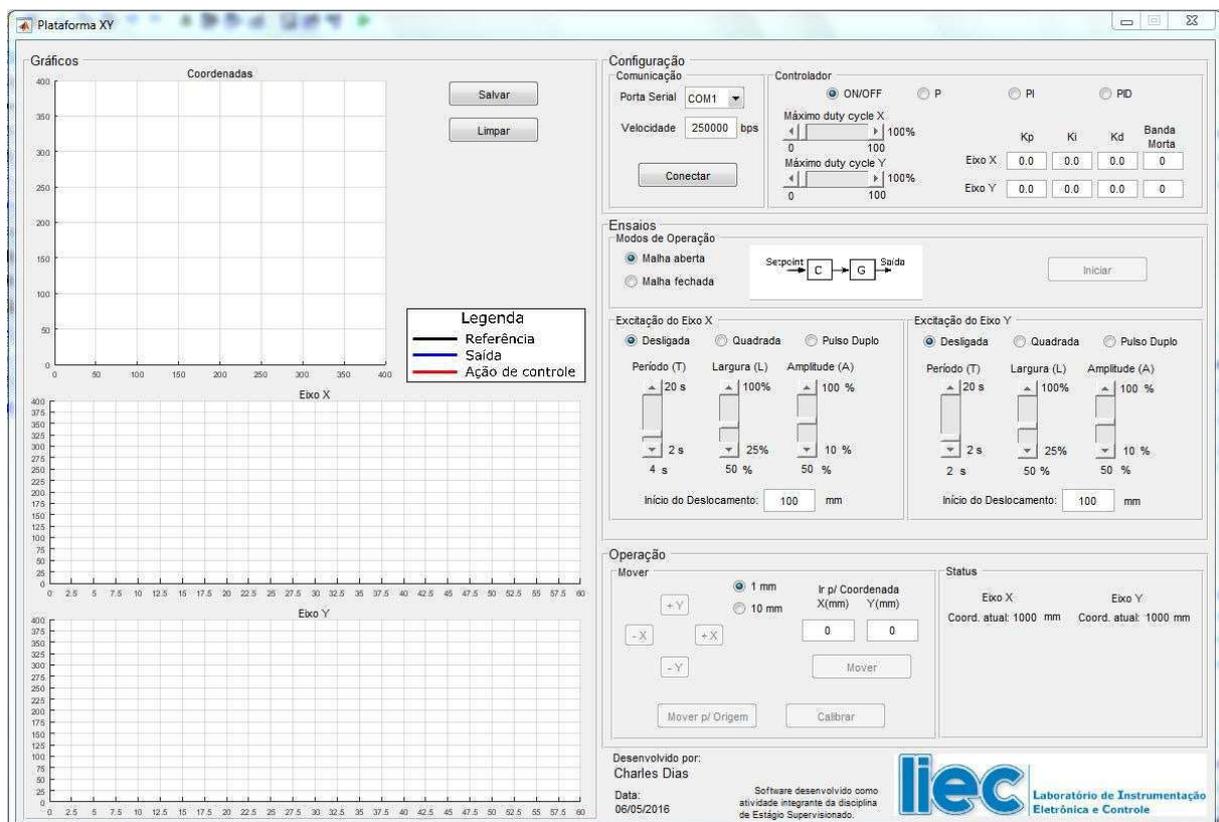


Fonte: Próprio autor.

Após a codificação dos casos de uso apresentados, foi obtido o programa denominado Plataforma XY, sendo sua interface apresentada na Figura 24. Observa-se que os quatro diagramas de caso de uso, apresentados anteriormente, foram agrupados em quatro *Panels*, no qual cada *Panel* concentrou as ações correspondentes a cada diagrama de caso de uso.

A utilização das funções presente nesse programa fora detalhada nos tópicos seguintes, sendo seu uso mostrado em detalhes.

Figura 24 – Interface do programa Plataforma XY.



Fonte: Próprio autor.

4 Plataforma XY

4.1 Descrição das funções da interface

O programa Plataforma XY conta com quatro *Panels*: Configuração, Ensaios, Operação e Gráficos. O objetivo de cada uma delas e como utilizá-las estão descritos a seguir.

4.1.1 Configuração

Na *Panel*, apresentada na Figura 25, constam as configurações referentes a comunicação serial e ao tipo de controlador utilizado.

Comunicação Serial

Para a comunicação serial, são disponibilizados os seguintes campos:

- Porta Serial – selecionar a porta serial ao qual está conectada a planta didática. Neste caso, a planta didática é conectada mediante uma porta USB, porém, será emulada uma serial virtual;
- Velocidade de comunicação – configura a velocidade de comunicação (baud rate). Apesar da possibilidade de alteração desse parâmetro, salienta-se que a planta está configurada para comunicação em 250000 bps.

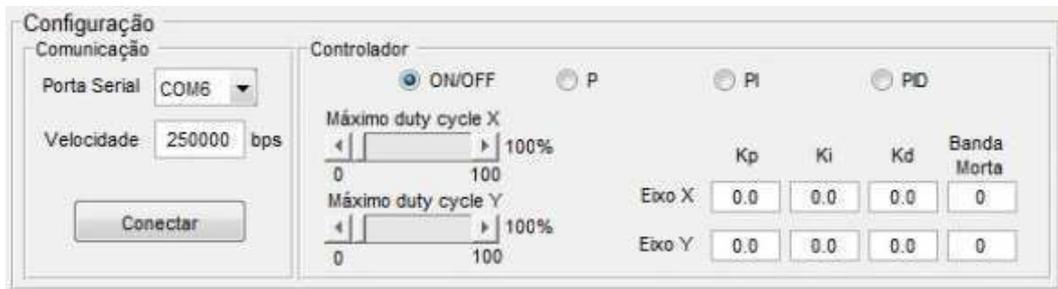
Controlador

Descrição dos campos para configuração do controlador:

- Tipo de controlador – são disponibilizados quatro tipos de controlador, sendo que a seleção é válida para ambos os eixos. Ao se conectar com a planta didática, o controlador On/Off é automaticamente selecionado;
- Máximo duty cycle – este parâmetro é válido apenas para o tipo de controlador ON/OFF. Este parâmetro configura a velocidade de deslocamento para o controlador citado, com possibilidade de configurar diferentes velocidades para os eixos;
- Ganhos – quando se seleciona o controlador P, PI ou PID, é possível configurar os ganhos desses controladores de maneira individual para cada eixo. Ao se conectar com a planta didática, os valores padrões dos ganhos são carregados;
- Banda morta – existe uma faixa de valor do *duty cycle*, que ao ser aplicado ao motor não produz movimento. Essa faixa de valores é chamada de banda

morta. Devido as diferenças construtivas dos eixos, tem-se uma pequena diferença entra a faixa de banda morta, sendo seu valor de aproximadamente 12%.

Figura 25 – *Panel* Configuração.



Fonte: Próprio autor.

4.1.2 Ensaios

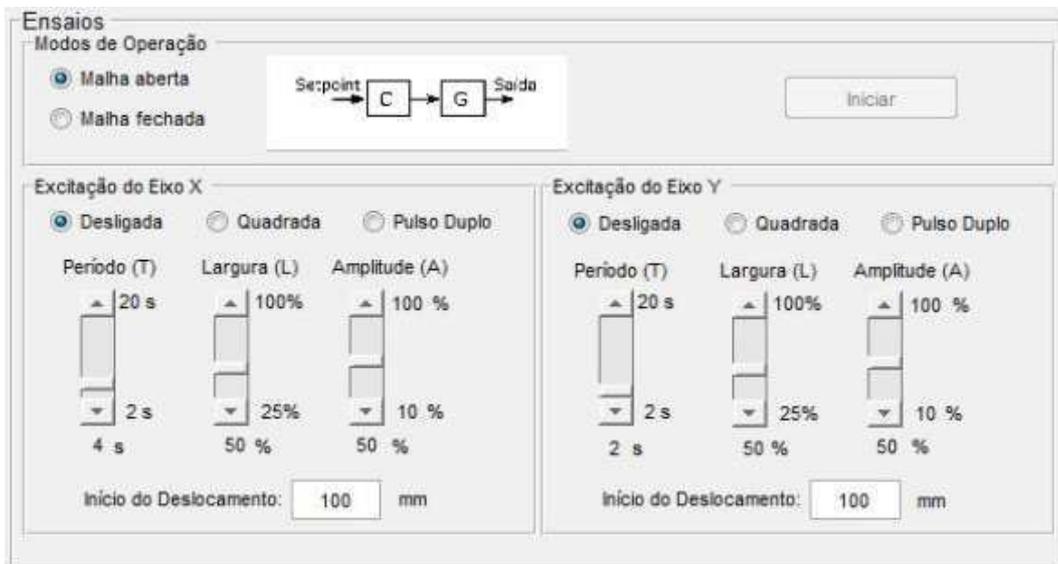
O programa Plataforma XY possibilita realizar ensaios em malha aberta ou em malha fechada. Os ensaios realizados em malha aberta, visam a aquisição dos dados necessário para implementação de técnicas de identificação e controle. Já os ensaios realizados em malha fechada, possibilitam verificar o desempenho do controlador escolhido. Na Figura 26, são exibidas as funções existente na *Panel* Ensaios, contendo as funções:

- Modos de operação – seleciona o tipo de ensaio, se em malha aberta ou malha fechada;
- Excitação do eixo – neste campo são encontrados parâmetros para configuração do tipo de onda que será utilizada para o ensaio em malha aberta ou em malha fechada. Observa-se que é possível selecionar diferentes parâmetros para os eixos;
 - Tipos de onda – são disponibilizados dois tipos de onda, quadrada ou pulso duplo, para excitar o eixo ou servir como sinal de referência;
 - Período (T) – ajusta o tempo de período da onda, sendo sua faixa compreendida entre 2 segundos e 20 segundos, com incrementos de 2 segundos;
 - Largura (L) – este parâmetro é utilizado apenas para a forma de onda pulso duplo, determinando o percentual de largura do pulso;
 - Amplitude (A) – estando selecionado o ensaio em malha aberta, esse valor corresponderá ao valor de excitação aplicado ao motor, o *duty*

cycle. Já em malha fechada, será o valor referente ao deslocamento da onda de referência;

- Início do deslocamento – trata-se da coordenada na qual o eixo irá se posicionar antes de iniciar o ensaio.

Figura 26 – *Panel* Ensaios.



Fonte: Próprio autor.

4.1.3 Operação

Deslocamento para coordenadas específicas, podem ser realizadas por meio da *Panel* Operação, exibida na Figura 27. Nela pode-se realizar algumas ações, tais como: deslocamento da plataforma para a origem; calibração, que atualiza o valor da variável pulso/milimeter; realizar incremento ou decremento nas coordenadas; e ainda, deslocar para um coordenada desejada.

Figura 27 – *Panel* Operação.

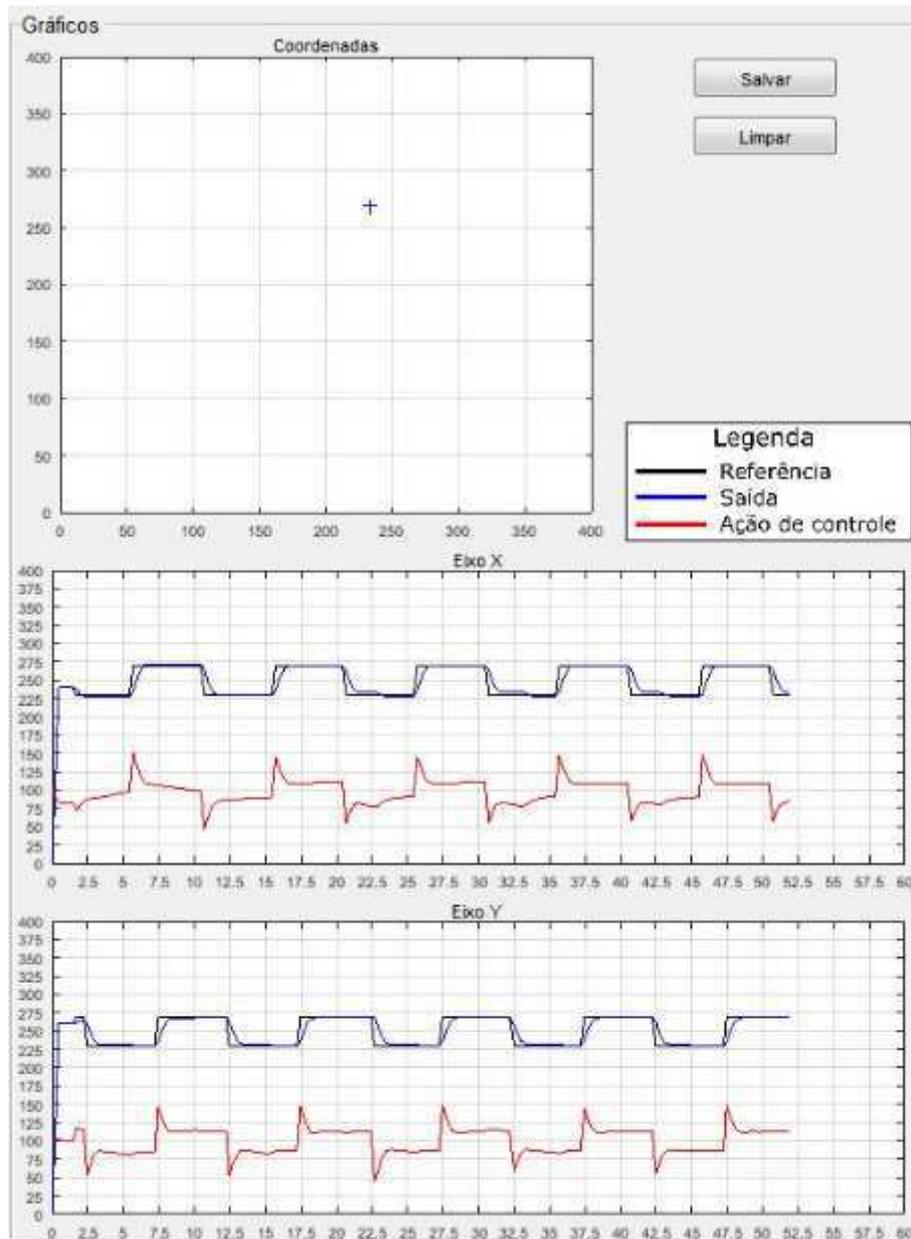


Fonte: Próprio autor.

4.1.4 Gráficos

Na *Panel* Gráficos, apresentada na Figura 28, são disponibilizados três gráficos que exibem informações como coordenada atual dos eixos e os sinais de referência, ação de controle e saída, para cada eixo. Ainda conta com a presença de dois botões: o Salvar, que armazena os dados em duas variáveis, dadosPlotX.mat e dadosPlotY.mat; e o de Limpar, que apaga os dados dessas variáveis e dos gráficos.

Figura 28 - *Panel* Gráficos.



Fonte: Próprio autor.

4.2 Identificação e controle da planta didática

Para que a sintonia do PID possa ser realizada, faz-se necessários que um modelo aproximado que descreva a dinâmica da planta seja determinado. Existem algumas técnicas que podem auxiliar na determinação desse modelo, como por exemplo, Ziegler-Nichols ou por modelagem matemática utilizando equações diferenciais.

Modelar matematicamente um processo pode ser uma tarefa demorada e muito complexa, pois necessita que se conheçam as equações matemáticas que descrevem o funcionamento do processo. No próximo tópico é realizada uma modelagem matemática de partes da planta didática, apenas com o objetivo de descrever a dinâmica dessas partes.

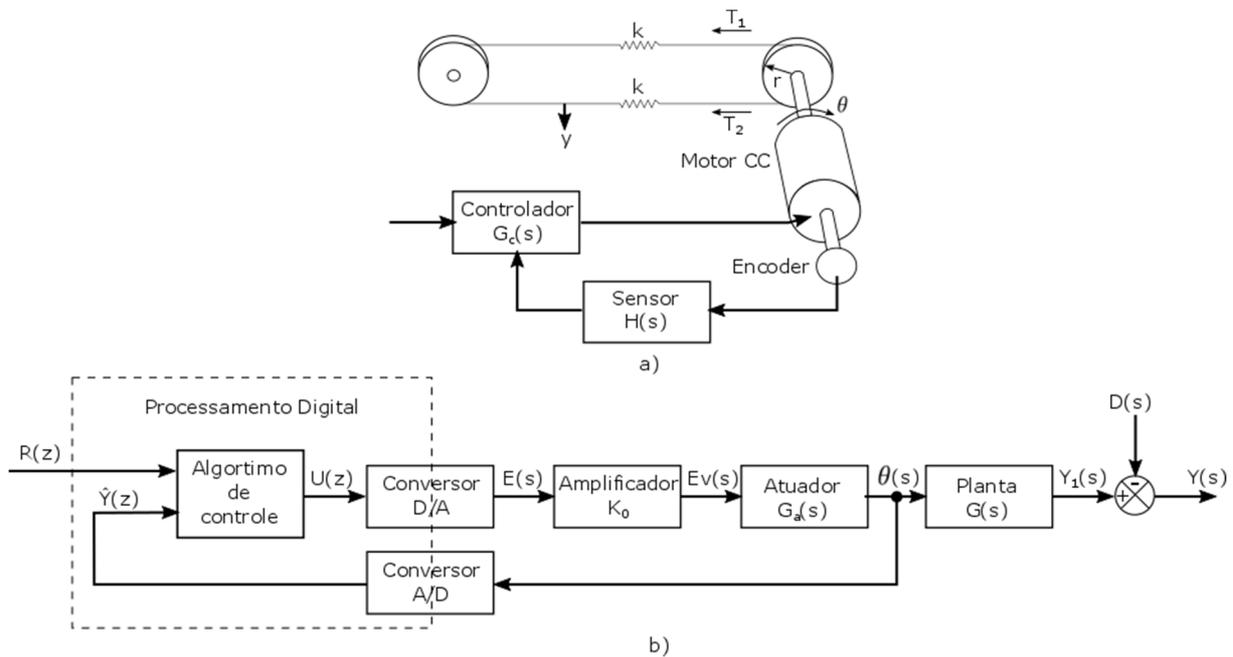
Dessa forma, faz-se necessário utilizar técnicas ou ferramentas que possam agilizar o processo de sintonia do controlador, sem que haja necessidade de descrever o processo por um conjunto de equações diferenciais. Conforme está descrito no tópico 4.2.3 Identificação do processo, a identificação e sintonia dos controladores PID's, foram realizadas por intermédio de ferramentas computacionais.

4.2.1 Modelagem matemática

Um sistema eletromecânico semelhante, apresentado e modelado por (Dorf & Bishop, 2013), foi utilizado como base para a formulação da modelagem da planta. Na Figura 29.a), é apresentado o diagrama para apenas um sistema de eixo. Conforme já informado, os eixos são compostos pelo mesmo conjunto eletromecânico, sendo suas diferenças presente nas questões que tange as dimensões de construção e número de elementos de máquinas utilizados.

Na Figura 29.b), é apresentado o diagrama de blocos para o sistema eletromecânico responsável pela atuação de um dos eixos.

Figura 29- a) diagrama para o sistema eletromecânico para um eixo e b) diagrama de blocos representando o funcionamento do sistema.



Fonte: Adaptado de (Dorf & Bishop, 2013).

Algoritmo de controle

Foi desenvolvido o firmware do microcontrolador composto por dois tipos de controladores: o controlador de duas posições ou On/Off; e o controlador PID. Contudo, devido ao desenvolvimento do firmware ter sido realizado com base na orientação a objeto, o acréscimo de outros tipos de controladores ou técnicas de controle podem ser realizadas mediante a realização de: inserir o novo controlador no firmware; adicionar funções para leitura e escrita dos parâmetros desse novo controlador, por intermédio do protocolo MODBUS; adicionar os campos pertinentes na interface do programa.

I. Controlador de duas posições ou On/Off

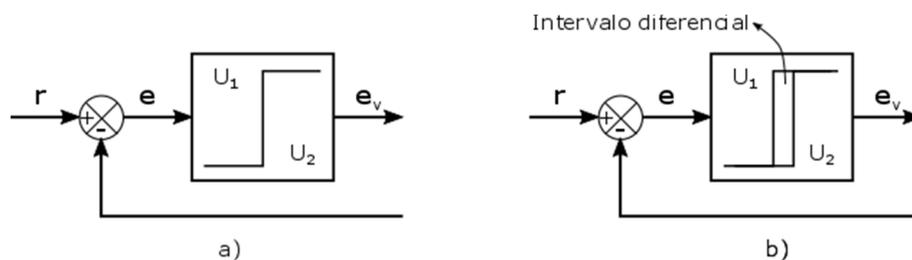
O controlador de duas posições ou On/Off possui apenas dois estados para o valor da sua saída, normalmente ligado (on) ou desligado (off). O controlador implementado possui uma pequena diferença, pois é possível configurar o valor máximo da tensão que alimenta o motor CC (atuador). A rotação no sentido horário equivalente ao estado *On*, e a rotação no sentido anti-horário, equivalente ao estado *Off*.

O valor do sinal $u(t)$ é dependente do sinal de erro atuante ser positivo ou negativo. Portanto,

$$u(t) = \begin{cases} U_1, \text{ para } e(t) > 0 \\ U_2, \text{ para } e(t) < 0 \end{cases}$$

sendo os valores de U_1 e U_2 correspondentes ao valor configurado no programa na opção *Configuração > Controlador > Máximo duty cycle*. A faixa de valores para a qual o erro atuante deve variar antes que ocorra a comutação, é chamada de intervalo diferencial. O diagrama de blocos desse controlador é apresentado na Figura 30, sendo o intervalo diferencial indicado na figura 30.b).

Figura 30 - Diagrama de blocos para o controlador On/Off.



Fonte: Adaptado do (Ogata, 2010).

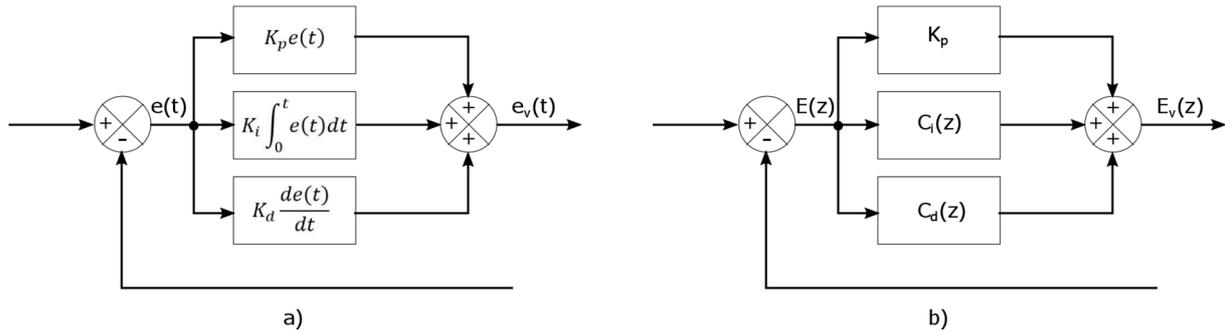
Espera-se que para pequenos valores de intervalo diferencial seja alcançado o valor do erro em regime permanente menor, o que é muito importante caso se deseje uma elevada precisão na posição. Entretanto, essa maior precisão tem como custo uma atuação mais frequente do atuador, o que além de aumentar o consumo de energia, também contribuirá para reduzir a sua vida útil.

O valor do intervalo diferencial foi incluído no controlador projetado por intermédio da variável *delta*. Este valor pode ser alterado diretamente no firmware (arquivo *CoordinateAxis.V03.h*), possibilitando realizar ensaios com outros valores de intervalos diferenciais. Este procedimento foi realizado e está comentando no tópico 4.2.5 Ensaio em malha fechada, página 43.

II. Controlador PID

A implementação do controlador PID foi realizada com a utilização do *PID Library*, que é uma biblioteca disponível para utilização com o Arduino. Na Figura 31 estão apresentados os diagramas de blocos para o PID em tempo contínuo e em tempo discreto.

Figura 31 – a) PID em tempo contínuo e b) PID em tempo discreto.



Fonte: Próprio autor.

O controlador PID digital, pode ser representado pela função de transferência apresentada na equação (4.1).

$$\frac{E_v(z)}{E(z)} = C(z) = K_p + C_i(z) + C_d(z) \quad (4.1)$$

Dependendo da aproximação utilizada, as funções $C_i(z)$ e $C_d(z)$ podem assumir valores diferentes. Numa aproximação bilinear, tem-se que os termos T_i e T_d são os tempos integral e derivativo, tais que $K_i = \frac{K_p}{T_i}$ e $K_d = K_p T_d$. E ainda que

$$C_i(z) = K_i \frac{T z + 1}{2 z - 1} = K_p \frac{T}{2 T_i} \frac{z + 1}{z - 1} = K_p \frac{1}{2 \tau_i} \frac{z + 1}{z - 1}$$

$$C_d(z) = K_d \frac{2 z - 1}{T z + 1} = K_d T_d \frac{2 z - 1}{T z + 1} = K_d \tau_d 2 \frac{z - 1}{z + 1}$$

Após algumas manipulações matemáticas, chega-se ao seguinte resultado:

$$C(z) = K_p \left(1 + \frac{1}{2 \tau_i} + 2 \tau_d \right) \frac{z^2 + \frac{\frac{1}{\tau_i} - 4 \tau_d}{1 + \frac{1}{2 \tau_i} + 2 \tau_d} z + \frac{\frac{1}{2 \tau_i} + 2 \tau_d - 1}{1 + \frac{1}{2 \tau_i} + 2 \tau_d}}{(z + 1)(z - 1)} \quad (4.2)$$

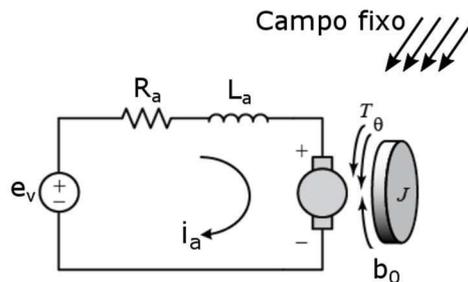
A equação (4.2) apresentada acima, teve como objetivo demonstrar a estrutura de um PID discreto, não sendo garantido que a aproximação utilizada na implementação do *PID Library* seguiu o mesmo modelo, uma vez que não foram encontradas informações sobre de como foi implementada essa biblioteca.

Os valores dos ganhos K_p , K_i e K_d , podem ser definidos mediante a configuração desses parâmetros na opção *Configuração > Controlador*.

Atuador Ga(s)

O atuador corresponde a um motor CC controlado pela corrente de armadura, uma vez que, nestes pequenos motores, são utilizados ímãs permanentes para gerar o fluxo magnético do campo. O diagrama de funcionamento para o motor CC é exibido na Figura 32.

Figura 32 – Diagrama para motor CC.



Fonte: <http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed§ion=SimulinkModeling>.

Para um fluxo de campo fixo, o torque desenvolvido pelo motor é dado por:

$$T = K_1 i_a \quad (4.3)$$

sendo K_1 a constante de torque do motor e i_a a corrente de armadura. Estando o eixo do motor em rotação, a tensão induzida na armadura e_b é diretamente proporcional à velocidade angular, isto é

$$e_b = K_2 \frac{d\theta}{dt} \quad (4.4)$$

sendo e_b a fcem (força contra eletromotriz), K_2 a constante de fcem do motor e θ o deslocamento angular do eixo do motor.

Aplicando a lei de Kirchhoff para a malha de tensão apresentada na Figura 32, tem-se a equação ((4.5).

$$e_v = L_a \frac{di_a}{dt} + R_a i_a + e_b = L_a \frac{di_a}{dt} + R_a i_a + K_2 \frac{d\theta}{dt} \quad (4.5)$$

Sabendo que a equação de equilíbrio do torque é:

$$J_0 = \frac{d^2\theta}{dt^2} + b_0 \frac{d\theta}{dt} = T = K_1 i_a \quad (4.6)$$

no qual, J_0 é o momento de inércia da combinação motor, carga e conjunto de engrenagem e b_0 o coeficiente de atrito viscoso, pode-se manipular as equações (4.5) e (4.6), obtendo assim, a função de transferência que da posição angular pela tensão de alimentação da armadura.

$$\frac{\theta(s)}{E(s)} = \frac{K_0 K_1}{s(L_a s + R_a)(J_0 s + b_0) + K_1 K_2 s} \quad (4.7)$$

sendo,

L_a = Indutância de armadura

R_a = Resistência de armadura

J_0 = Momento de inércia

b_0 = Conjunto de atritos

K_0 = fator de amplificação entre $E(s)$ e $E_v(s)$

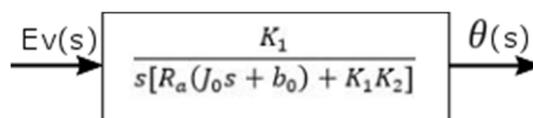
K_1 = Constante de torque do motor

K_2 = Constante de fcm do motor

Para motores de pequena potência, o valor de $L_a \ll R_a$, podendo ser realizada uma simplificação na equação (4.7), o que leva a equação (4.8) e ao diagrama de bloco representado na Figura 33.

$$\frac{\theta(s)}{E_v(s)} = \frac{K_1}{s[R_a(J_0 s + b_0) + K_1 K_2]} \quad (4.8)$$

Figura 33 – Diagrama de bloco para a função de transferência do motor CC.



Fonte: Próprio autor.

Planta $G(s)$

O eixo do motor está acoplado a um sistema de engrenagens (caixa de redução), cuja relação é n , aproximadamente 74,83:1. Uma polia de raio r está acoplada ao eixo do sistemas de engrenagem, transformando o movimento radial em movimento linear, mediante o deslocamento de uma correria sincronizadora. Assim, a função de transferência da planta pode ser modelada pela (4.9).

$$\frac{\theta(s)}{Y_1(s)} = \frac{2\pi r}{n} \quad (4.9)$$

Conversor A/D

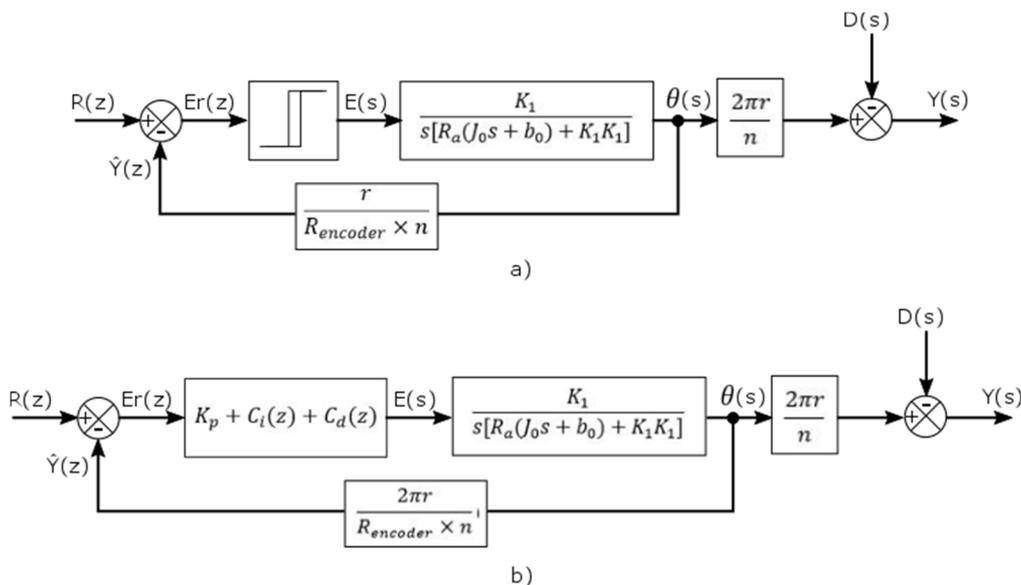
O conversor A/D representa o algoritmo que estima o valor da saída baseado na rotação do eixo do motor $\theta(s)$, no raio da polia r e na resolução do encoder $R_{encoder}$. Dessa forma, o valor estimado da saída $\hat{Y}(z)$ poderia ser modelado por:

$$\hat{Y}(z) = \frac{2\pi r}{R_{encoder} \times n} \theta(s) \quad (4.10)$$

Planta didática

Retomando o diagrama de blocos apresentado na Figura 29, e substituindo as funções de transferência calculadas, tem-se o diagrama apresentado na Figura 34.

Figura 34 – a) planta com controlador On/Off e b) planta com controlador PID discreto.



Fonte: Próprio autor.

O processo de modelagem matemática de um processo, pode se tornar uma tarefa complexa e demorada, necessitando assim que uma outra abordagem seja realizada. Dessa forma, a planta foi projetada de forma a possibilitar a realização de ensaios em malha aberta e em malha fechada, possibilitando a coleta dos dados gerados nesses ensaios, permitindo a utilização de outras técnicas de identificação do processo e de sintonia dos controladores.

4.2.2 Ensaio em malha aberta

O ensaio em malha aberta visa realizar a coleta de dados mediante a aplicação de um sinal conhecido à entrada da planta didática, seguido do monitoramento do

valor da saída do sistema. Esses dados são exibidos no gráfico, para o eixo correspondente, podendo ser salvos para posterior aplicação de métodos de identificação e controle, sejam eles com auxílio de ferramentas computacionais ou não.

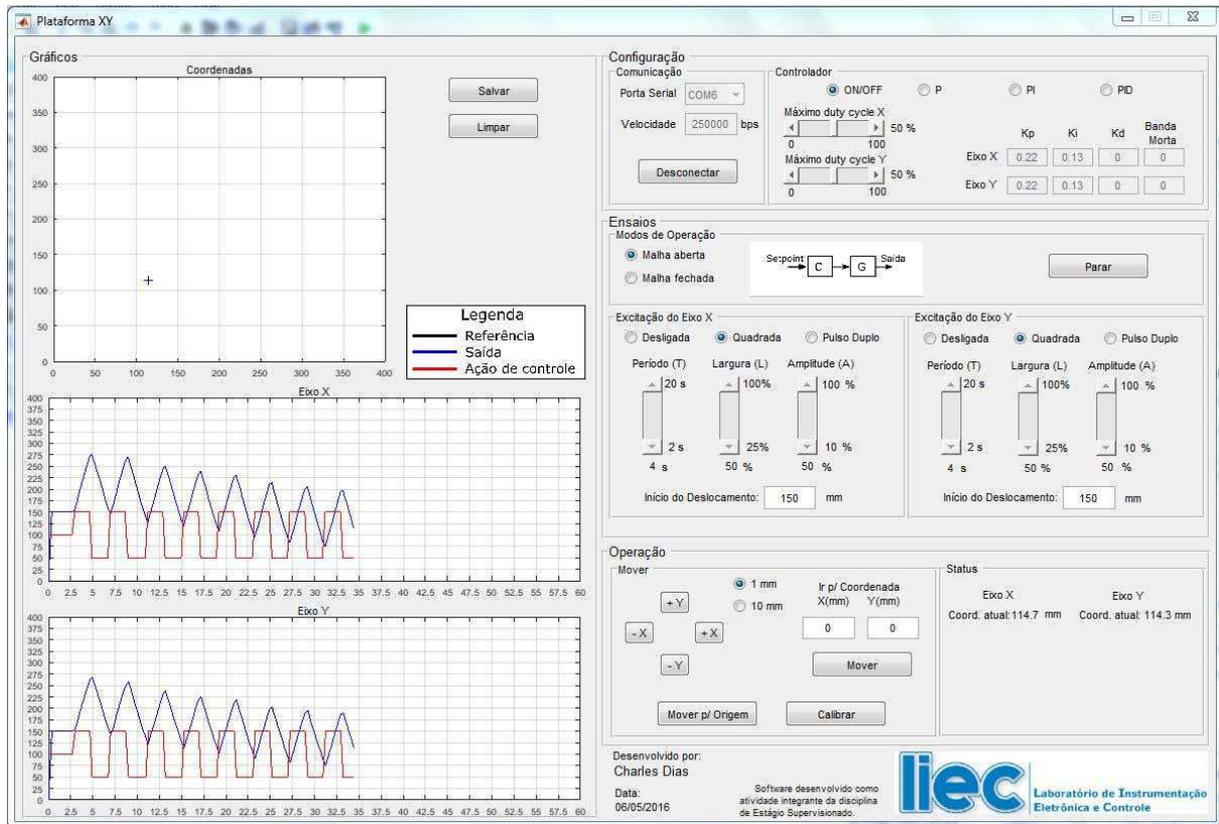
A seguir estão descritos os passos para realização do ensaio em malha aberta utilizando como sinal de excitação uma onda quadrada. Serão descritos apenas os procedimentos para o eixo X, sendo estendidos de maneira análoga ao eixo Y.

Realização do ensaio em malha aberta – Onda quadrada

1. Realizar a conexão entre a planta didática e o programa Plataforma XY;
2. Selecionar o modo de operação malha aberta, *Ensaio > Modos de Operação > Malha Aberta*.
3. Selecione a onda quadrada para excitação do eixo X, *Ensaio > Excitação do Eixo X > Quadrada*;
4. Configure a onda quadrada para o período de 4 segundos e amplitude de 50% para ambos os eixos;
5. Pressione no botão Iniciar.
6. Decorrido um certo tempo, o ensaio pode ser interrompido e os dados armazenados pressionando o botão Salvar.

Na Figura 35, é exibida a tela do programa com as configurações descritas anteriormente, na qual constam também os sinais de excitação e de saída para ambos os eixos.

Figura 35 - Ensaio em malha para ambos os eixos.



Fonte: Próprio autor.

4.2.3 Identificação do processo

A identificação da planta didática foi realizada com o auxílio da ferramenta computacional MATLAB, mais especificamente com a utilização da *System Identification Toolbox™*, que se trata de um aplicativo para a construção de modelos matemáticos de sistemas dinâmicos a partir dos dados medidos de entrada e saída.

Utilizando *System Identification*, os dados obtidos no ensaio em malha aberta foram importados para que as funções de transferências fossem estimadas. Entretanto, observa-se a existência de uma não linearidade na resposta em malha aberta de ambos os eixos. Assim, apenas um intervalo dos dados foi utilizado na identificação.

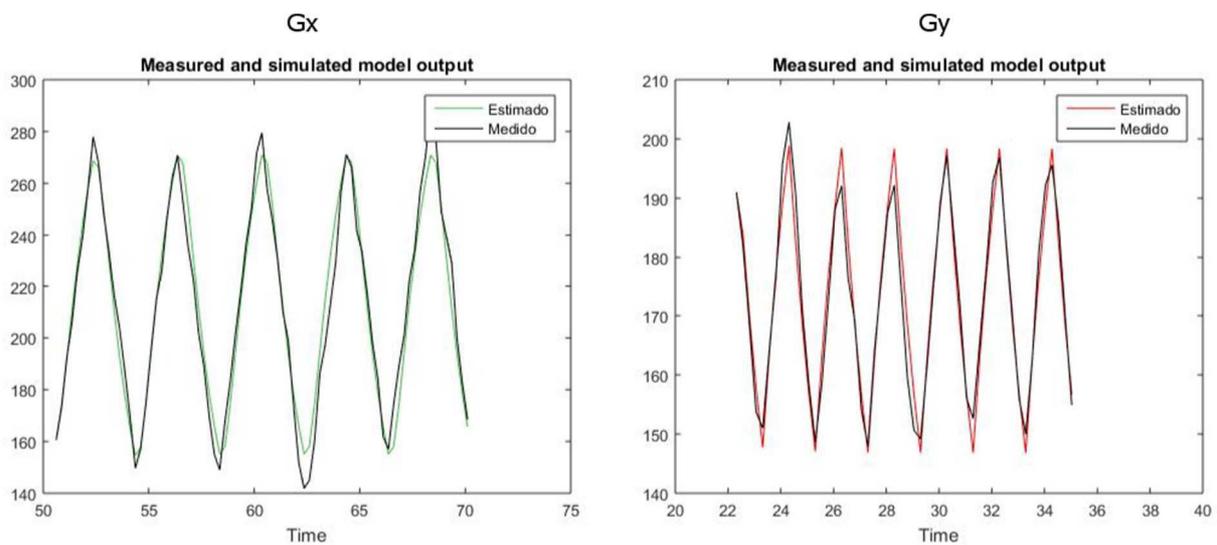
Sabendo que a função de transferência do atuador é um modelo aproximado que contém dois polos e nenhum zero, decidiu-se estimar o modelo do eixo seguindo essa mesma estrutura. Dessa forma, foram obtidas as funções de transferência para os eixos X e Y, equações (4.11) e (4.12).

$$G_x(s) = \frac{5,089}{s^2 + 3,777 s + 2,389} \quad (4.11)$$

$$G_y(s) = \frac{218.7}{s^2 + 182,4 s + 126,7} \quad (4.12)$$

Para o ponto de operação escolhido, as funções apresentadas nas equações acima foram consideradas válidas, pois segundo a estimativa do MATLAB tem-se uma semelhança de 78,42% e 77,52% para o intervalo de dados experimentais utilizados, conforme gráficos apresentados na Figura 36.

Figura 36 – Comparativo entre os sinais estimados e medidos.



Fonte: Próprio autor.

4.2.4 Sintonia dos controladores

Observando as funções de transferências que foram obtidas, percebe-se que os eixos X e Y apresentam dinâmicas diferentes. Essa constatação já era esperada, devido as diferenças construtivas que acarretam um conjugado mecânico diferente para cada arranjo.

Como os eixos apresentam dinâmicas diferentes, fez-se necessário realizar a sintonia para ambos os controladores PID. Assim como foi realizado no processo de identificação, a sintonia dos controladores contou com a utilização da ferramenta computacional MATLAB, ferramenta *PID Tuner*.

Os parâmetros foram determinados de forma que as respostas dos sistemas não apresentassem sobressinais e uma resposta suave do controlador, objetivando

evitar ainda, a saturação do controlador. Dessa forma, os ganhos resultantes foram calculados, sendo exibidos na Tabela 5.

Tabela 5 - Ganhos dos controladores PID para os eixos.

Eixo	Kp	Ki	Kd
X	0,15	0,09	0
Y	0,17	0,12	0

Fonte: Próprio autor.

4.2.5 Ensaio em malha fechada

Controlador On/Off

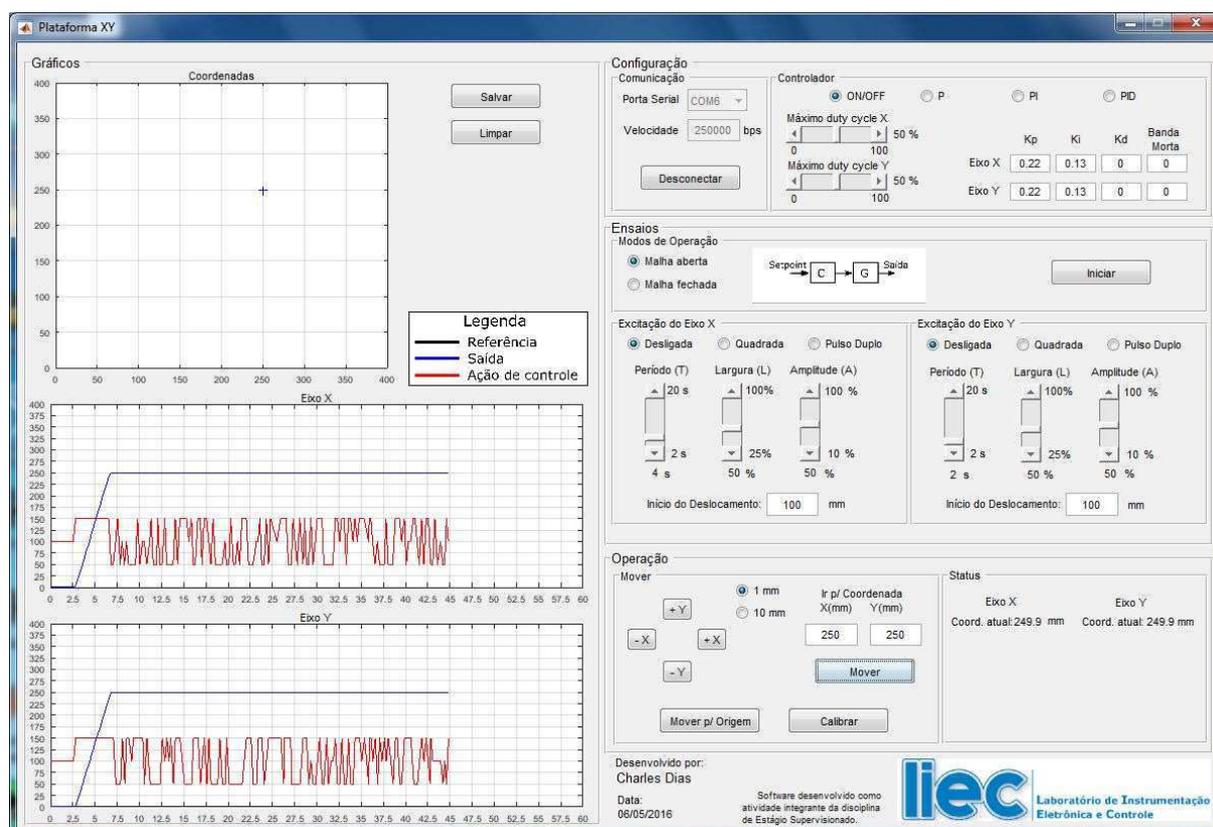
Foram realizados dois ensaios em malha fechada utilizando o controlador On/Off: para o primeiro, o valor da variável delta, que corresponde ao intervalo diferencial, foi configurado para 0,02; para o segundo, delta foi igual a 0,2. É importante lembrar que a alteração do valor da variável delta só pode ser realizada diretamente no firmware.

Para ambos os valores de delta, foram realizados o ensaio em malha fechada, seguindo os procedimentos de configuração descritos abaixo:

- Selecionar o controlador On/Off na opção *Configurações > Controlador*;
- Selecionar o modo de operação em malha fechada;
- Selecionar a excitação do eixo X e do eixo Y como uma onda quadrada;
- Período da onda para 10 segundo e amplitude de 50 mm.

Na Figura 37, tem-se a imagem da tela do programa, na qual podem ser visualizadas os sinais de ação de controle e da saída, quando é aplicado um degrau de 250 mm e valor de delta = 0,02.

Figura 37 – Ensaio em malha fechada utilizando o controlador On/Off.



Fonte: Próprio autor.

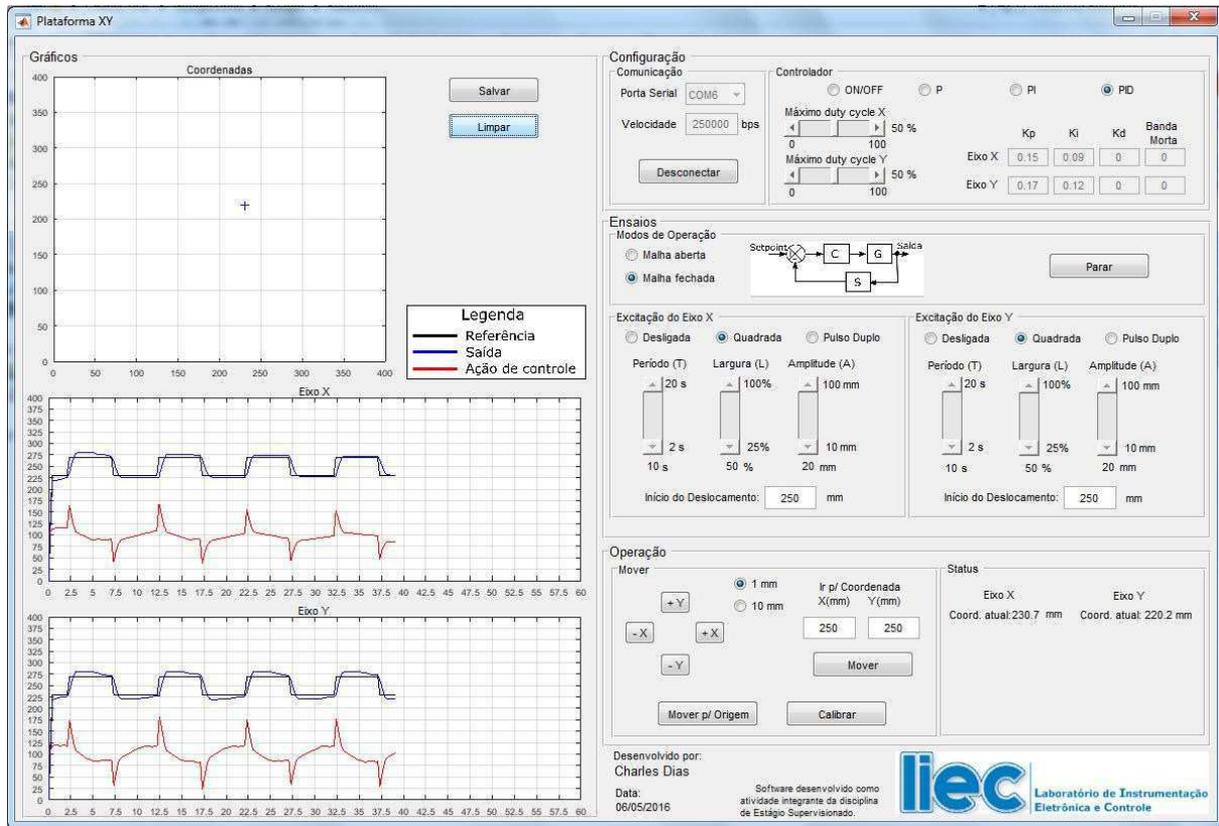
Controlador PID

Para verificar o desempenho dos controladores PIDs, foram realizados ensaios em malha fechada para os dois eixos. Os procedimentos de configuração para realização dos ensaios são descritos abaixo:

- Inserir os ganhos calculados na em *Configurações > Controlador*;
- Selecionar o modo de operação em malha fechada;
- Selecionar a excitação do eixo X e do eixo Y como uma onda quadrada;
- Período da onda para 10 segundo e amplitude de 20 mm.

Na Figura 38, tem-se a imagem da tela do programa, na qual podem ser visualizadas os sinais de referência, ação de controle e saída, para ambos os eixos.

Figura 38 – Ensaio em malha fechada para os ganhos mostrados na Tabela 5.



Fonte: Próprio autor.

4.3 Resultados

Controlador On/Off

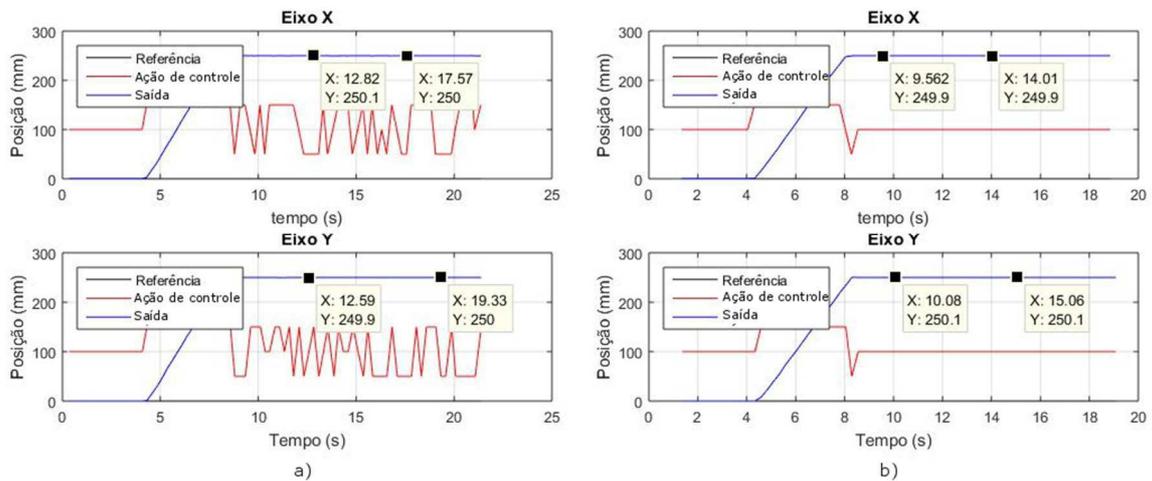
Na Figura 39, estão apresentadas as respostas do controlador On/Off quando é aplicado um degrau de 250 mm.

Observa-se que para um intervalo diferencial de 0,02, Figura 39.a), tem-se uma constante atuação da ação de controle, pois o valor da saída se encontra fora do intervalo diferencial. Essa constante atuação acarreta no aumento do consumo de energia e contribui para redução da vida útil do atuador. Já na Figura 39.b), foi aplicado o mesmo degrau de 250 mm, porém, para um valor de intervalo diferencial de 0,2. Para essa segunda situação, o atuador é acionado por um tempo bem inferior, uma vez que a saída permanece dentro do intervalo diferencial.

Outro ensaio em malha fechada foi realizado, sendo utilizado uma onda quadrada como sinal de referência, Figura 40. Analisando o gráfico, chega-se à conclusão que a saída segue o sinal de referência praticamente sem sobressinal e com um tempo de acomodação da ordem de 2 segundos. Porém, cabe uma ressalva

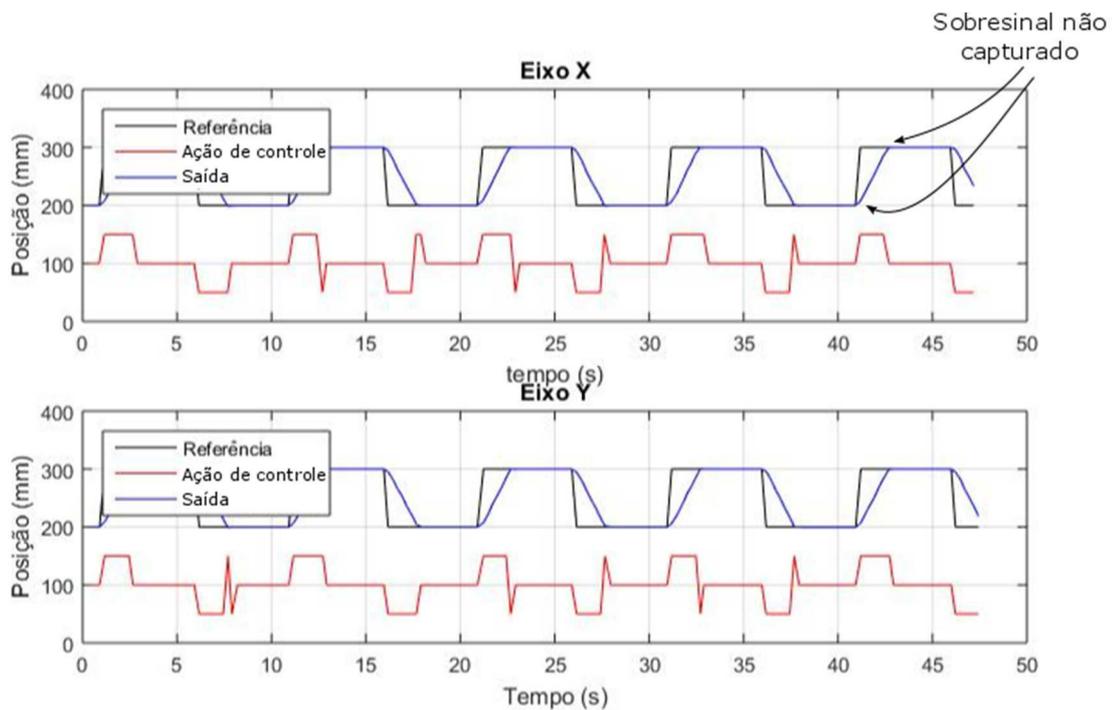
aqui. Existe sobressinal na saída, entretanto, devido ao tempo de aquisição dos sinais e a rápida ação do controlador On/Off, esse sobressinal não é mostrado no gráfico.

Figura 39 - Sinais de referência, ação de controle e saída para o ensaio em malha fechada utilizando controlador On/Off. a) Delta = 0,02 mm e b) Delta = 0,2 mm.



Fonte: Próprio autor.

Figura 40 - Utilização do controlador On/Off para seguir uma onda quadrada como sinal de referência.



Fonte: Próprio autor.

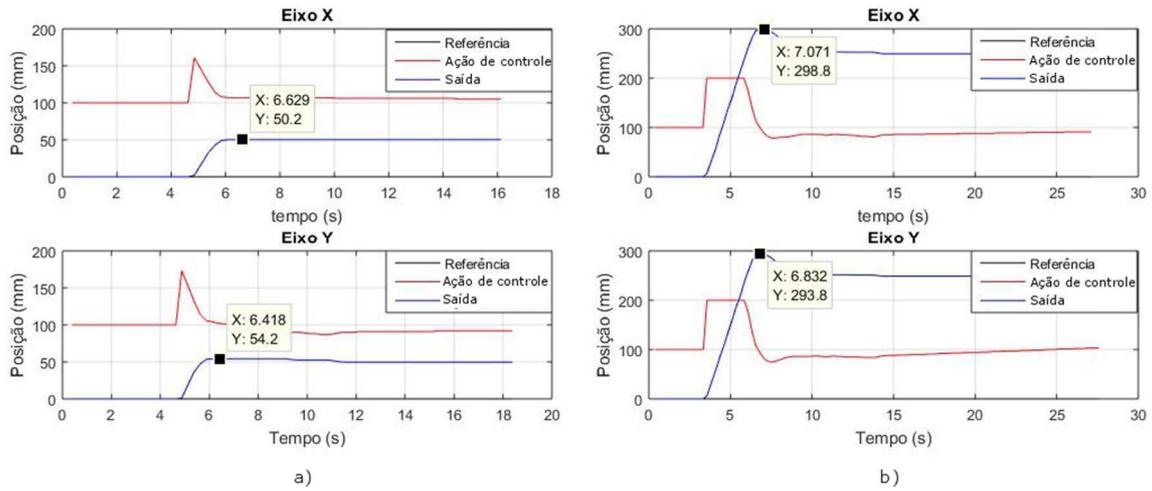
Controlador PID

Na Figura 41, estão apresentadas as respostas para um degrau de 50 mm e de 250 mm. Observa-se que o degrau de 50 mm, a saída apresentou um sobressinal de 8%, aproximadamente, e um tempo de acomodação de 1 segundo. Já para o degrau de 250 mm, o sobressinal foi de 20% e o tempo de acomodação de 3,5 segundos, aproximadamente.

O ensaio em malha fechada com a utilização de uma onda quadrada como sinal de referência, é exibido na Figura 42. Para esse ensaio, o deslocamento é de 20 mm em torno do ponto de operação 250 mm. Verifica-se que a saída segue o sinal de referência, sem sobressinal e com um tempo de acomodação da ordem de 1 segundo.

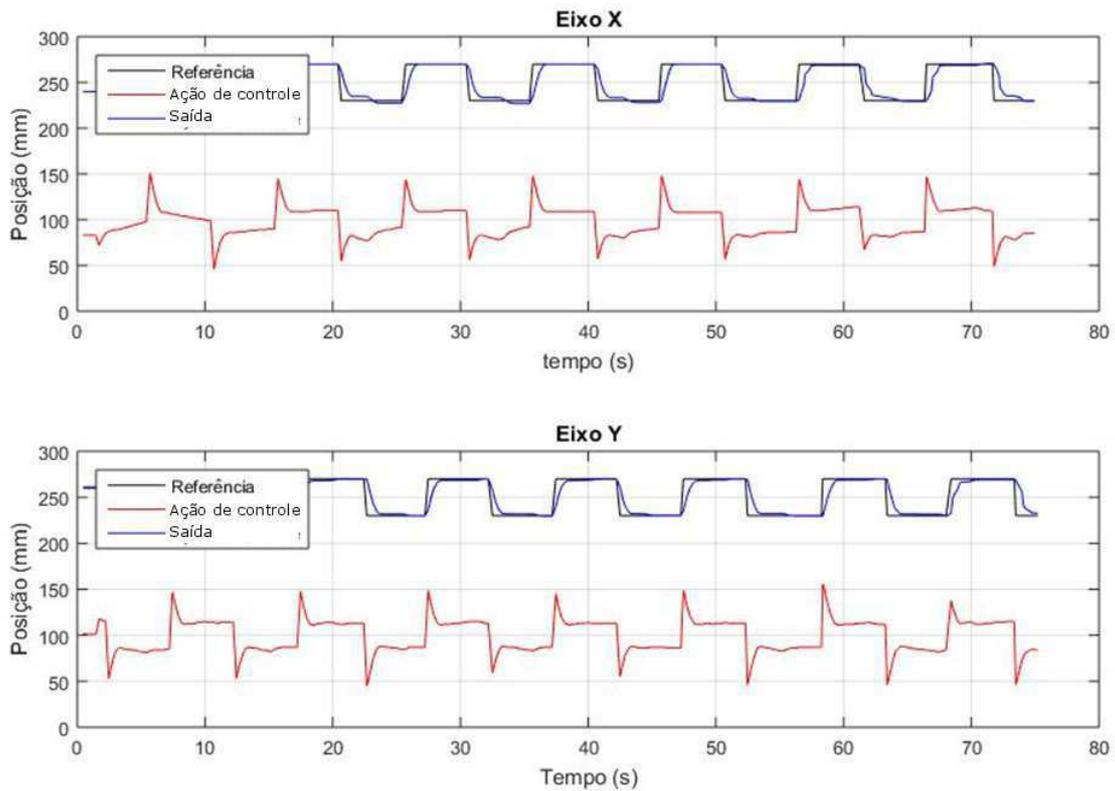
Conclui-se que o controlador PID funciona de forma a atender as especificações, apenas para deslocamentos pequenos, 20 mm ou inferior. Para deslocamento maiores, começa a existir um valor significativo de sobressinal. Esse fato se deve a não linearidade apresentada pelo sistema, decorrente principalmente do maior momento de inércia adquirido pelo conjunto mecânico, quando realiza deslocamentos maiores.

Figura 41 - Sinais de referência, ação de controle e saída para o ensaio em malha fechada utilizando controlador PID. a) Degrau de 50 mm e b) Degrau de 250 mm.



Fonte: Próprio autor.

Figura 42 – Utilização do controlador PID para seguir uma onda quadrada como sinal de referência.



Fonte: Próprio autor.

4.4 Melhorias sugeridas

Com a conclusão da planta didática e do programa Plataforma XY, diversos testes foram realizados e foram constatados alguns pontos que poderiam ser melhorados. Esses pontos estão listados abaixo:

1. Aumentar a densidade de dados na comunicação entre a planta e o computador. Para isso, pode ser implementada a função do MODBUS que permite a operação de escrita e leitura em múltiplos registradores. Ou mesmo, o desenvolvimento de um protocolo com maior densidade de dados;
2. Eliminar as folgas do conjunto mecânico, possibilitando um deslocamento uniforme;
3. Pesquisar outros tipos de controladores que pudesse responder melhor à não-linearidade da planta;
4. Adicionar outros sinais para excitação dos eixos;
5. Possibilitar a entrada de um conjunto de coordenadas como referência.

5 Conclusão

O desenvolvimento da solução solicitada, que teve como objetivo realizar a montagem, identificação e controle de um posicionador de dois eixos para ser utilizado como plataforma didática, apresentou um conjunto de desafios, nos quais, os conhecimentos adquiridos em diversas disciplinas foram necessários para que o objetivo fosse alcançado.

É possível associar o êxito na execução das atividades, mais especificamente, como algumas disciplinas do curso: Introdução à Programação e Técnicas de Programação, para elaboração do programa embarcado no Arduino e o desenvolvimento do programa no ambiente MATLAB; Dispositivos Eletrônicos e Eletrônica, para desenvolvimento de circuitos eletrônicos e escolhas dos módulos, driver de potência e contador de pulso; Arquitetura de Sistemas Digitais para programação do Arduino e o desenvolvimento do firmware; Controle Analógico e Controle Digital no processo de identificação e controle da planta; e Automação Industrial que contribuiu para definir a estrutura de funcionamento do programa e o protocolo de comunicação.

Por fim, é certo que a realização desse estágio contribuiu sobremaneira para a formação profissional do estagiário, uma vez que os desafios apresentados, além de possibilitar uma vasta aplicação dos conhecimentos adquiridos, ainda ofereceu a oportunidade da elaboração completa de solução para um problema real de engenharia.

6 Bibliografia

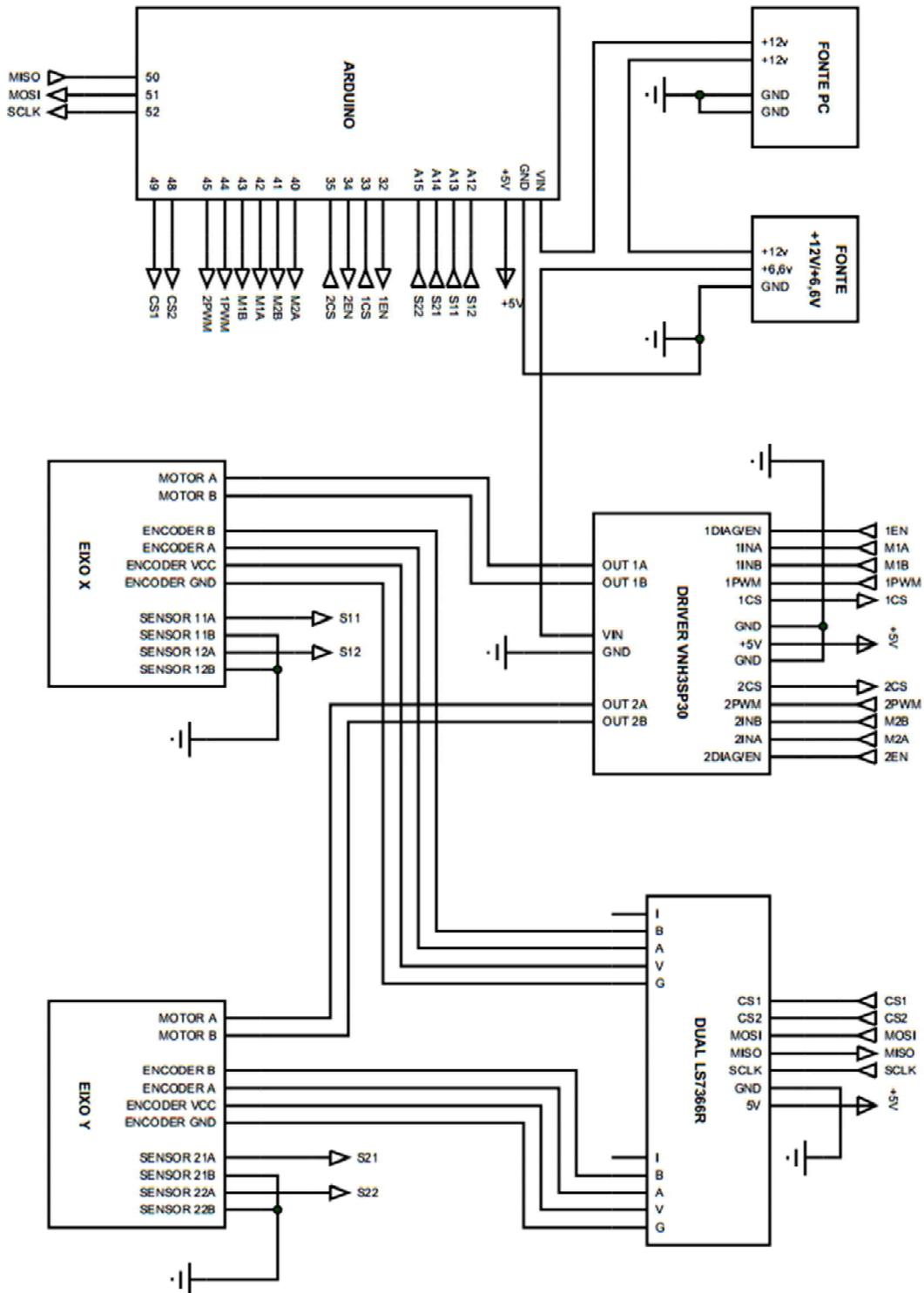
Dorf, R. C. & Bishop, R. H., 2013. *Sistemas de Controle Modernos*. 12ª ed. Rio de Janeiro: LTC.

LIEC, 2015. *Laboratório de Instrumentação Eletrônica e Controle*. [Online] Available at: <http://liec.ufcg.edu.br/> [Acesso em 11 05 2016].

Ogata, K., 2010. *Engenharia de Controle Moderno*. 5ª ed. São Paulo: Pearson.

APÊNDICE

Figura 43 – Circuito elétrico da planta.



Fonte: Próprio autor.