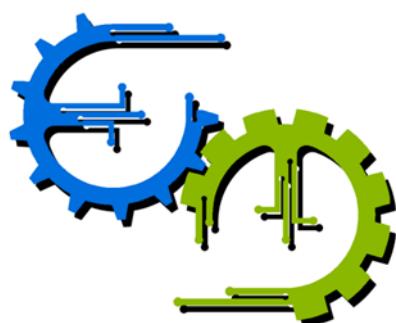


TRABALHO DE GRADUAÇÃO

**PROJETO E IMPLEMENTAÇÃO EM CONTROLADOR
INDUSTRIAL PARA POSICIONAMENTO DE RISERS
COM VALIDAÇÃO EXPERIMENTAL**

Por,
Ataias Pereira Reis
Emanuel Pereira Barroso Neto

Brasília, novembro de 2015



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

PROJETO E IMPLEMENTAÇÃO EM CONTROLADOR INDUSTRIAL PARA POSICIONAMENTO DE RISERS COM VALIDAÇÃO EXPERIMENTAL

Por,
Ataias Pereira Reis
Emanuel Pereira Barroso Neto

*Relatório submetido como requisito parcial de obtenção
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Eugênio Libório Feitosa Fortaleza, _____
ENM/UnB
Orientador

Prof. Eduardo Stockler Tognetti, ENE/UnB _____
Co-orientador

Brasília, novembro de 2015

FICHA CATALOGRÁFICA

REIS, ATAIAS PEREIRA; NETO, EMANUEL PEREIRA BARROSO;
Projeto e Implementação em Controlador Industrial para Posicionamento de Risers com Validação Experimental ,
[Distrito Federal] 2015.
x, 47p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2015). Trabalho de Graduação – Universidade de Brasília.Faculdade de Tecnologia.

1. Controle em Malha Fechada 2. Controlador Lógico Programável
3. Sistemas Offshore
I. Mecatrônica/FT/UnB

REFERÊNCIA BIBLIOGRÁFICA

REIS, A. P., NETO, E. P. B. (2015). Projeto e Implementação em Controlador Industrial para Posicionamento de Risers com Validação Experimental. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº0XX, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 109p.

CESSÃO DE DIREITOS

AUTORES: Ataias Pereira Reis e Emanuel Pereira Barroso Neto

TÍTULO DO TRABALHO DE GRADUAÇÃO: Projeto e Implementação em Controlador Industrial para Posicionamento de Risers com Validação Experimental.

GRAU: Engenheiro

ANO: 2015

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito dos autores.

Ataias Pereira Reis

72926-048 Águas Lindas – GO – Brasil

Emanuel Pereira Barroso Neto

72130-450 Taguatinga – DF – Brasil

Dedicatórias

A Emanuel B., meu avô, fonte de inspiração eterna

Emanuel Pereira Barroso Neto

Dedico à minha família, que me apoiou ao longo desta jornada.

Ataias Pereira Reis

Agradecimentos

Eu agradeço a Deus, aos professores Eugênio Fortaleza e Eduardo Tognetti, orientadores deste trabalho, e a todos que realizaram trabalhos prévios que foram utilizados aqui. Agradeço também ao sólcito Rédytton Brenner que nos ajudou muito no início.

Ataias Pereira Reis

Agradeço a Deus, aos professores Eugênio Fortaleza e Eduardo Tognetti, orientadores deste trabalho, sempre disponíveis quando mais precisamos; ao grande amigo Rédytton Brenner, que utilizou a bancada antes de nós e nos deu várias dicas úteis; a todos os meus amigos e familiares, que me deram forças nessa jornada; por fim, agradeço à minha dupla, Ataias, pela paciência, cordialidade e principalmente pela ajuda.

Emanuel Pereira Barroso Neto

RESUMO

A extração de petróleo em águas profundas requer operações bastante complexas. Em especial, a operação de entrada reentrada ocorre quando um *riser* é conectado de sua plataforma a um poço de petróleo. O deslocamento do *riser* até o poço deve ser feito de forma rápida e precisa, mas, atualmente, o controle de posição é manual, o que pode ser ineficiente. O presente trabalho busca validar técnicas de controle por meio de experimentos em uma planta de laboratório representativa de um *riser* por meio de controle em malha aberta e fechada para operar o deslocamento do *riser* de forma automática, com resultados rápidos e precisos, além de oferecer resistência a perturbações causadas pelo oceano, para que a ponta do *riser* seja corretamente conectada ao poço.

Palavras Chave: *riser*, controle, deslocamento

ABSTRACT

Deep sea petroleum exploration requires very complex operations. Particularly, the re-entry operation occurs when a riser is connected from the platform to a wellhead. The riser's displacement must be done quickly and precisely but, presently, the position control is manual, which may be inefficient. The present paper seeks to validate control techniques with experiments in a *riser* representative lab plant using open and closed loop control to operate the riser's displacement automatically, with fast and precise results, in addition to offering resistance against disturbances caused by the ocean, so that riser's tip is correctly connected to the wellhead.

Keywords: *riser*, control, displacement

SUMÁRIO

1	Introdução	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	OBJETIVOS DO PROJETO	1
2	Fundamentos	4
2.1	EQUAÇÕES GOVERNANTES	4
2.2	CONTROLE	4
2.3	BANCADA	5
2.3.1	CONTROLADOR LÓGICO-PROGRAMÁVEL	5
2.3.2	SERCOS INTERFACE	7
2.3.3	LINE INTERFACE MODULE 2094-AL09	8
2.3.4	DRIVE KINETIX 6000 DA ALLEN BRADLEY	8
2.3.5	SERVOMOTOR	8
2.3.6	SENSORES INDUTIVOS	9
2.3.7	CÂMERA PRESENCEPLUS	10
2.4	REDES UTILIZADAS	11
2.4.1	DEVICENET	11
2.4.2	ETHERNET/IP	11
2.5	PROGRAMAÇÃO DO CLP	12
2.5.1	VISÃO GERAL	12
2.5.2	LINGUAGEM <i>ladder</i>	12
2.5.3	TEXTO ESTRUTURADO	14
2.6	PARÂMETROS PARA GERAR TRAJETÓRIAS	15
3	Resultados	16
3.1	CÂMERA	16
3.1.1	CONFIGURAÇÃO DA REDE	16
3.1.2	CALIBRAÇÃO	17
3.1.3	PROGRAMAÇÃO	18
3.2	CALIBRAÇÃO DO SERVOMOTOR	18
3.3	MALHA ABERTA	19
3.3.1	EXCURSÃO DE 30CM	19
3.3.2	EXCURSÃO DE 20CM	20

3.4	MALHA FECHADA.....	21
3.4.1	CONTROLADOR P	22
4	Conclusões.....	23
4.1	PERSPECTIVAS FUTURAS.....	23
REFERÊNCIAS BIBLIOGRÁFICAS		24
Anexos.....		26
I	Programas utilizados.....	27
I.1	TEXTO ESTRUTURADO	27
I.1.1	INICIALIZAÇÃO DO PRIMEIRO TESTE DE MALHA ABERTA	27
I.1.2	INICIALIZAÇÃO DO SEGUNDO TESTE DE MALHA ABERTA.....	31
I.1.3	INICIALIZAÇÃO DOS TESTES DE MALHA FECHADA	33
I.1.4	INICIALIZAÇÃO DA REDE DEVICENET	34
I.1.5	PROGRAMA DE CONTROLE EM MALHA ABERTA	34
I.1.6	PROGRAMA DE CONTROLE PROPORCIONAL - MALHA FECHADA	34
I.2	LINGUAGEM <i>ladder</i>	35
I.2.1	EXECUÇÃO DE <i>trigger</i> DA CÂMERA	35
I.2.2	ROTINA DE PARADA DE EMERGÊNCIA	35
I.3	PROGRAMAS DA CÂMERA	35
I.3.1	DETECÇÃO DA POSIÇÃO HORIZONTAL DA BOLINHA.....	35

LISTA DE FIGURAS

1.1	Operação de reentrada [1]	2
1.2	Método atual para reconexão no poço [2]	2
1.3	Método proposto para reconexão no poço [2]	2
2.1	Malh.a aberta de controle	5
2.2	Malha fechada de controle.....	5
2.3	Esquemático da Bancada Utilizada para o Experimento [2]	6
2.4	CLP com identificação de elementos.....	6
2.5	Line Interface Module modelo 2094-AL09 da Allen Bradley [2]	8
2.6	Drive Kinetix 6000 da Allen Bradley	9
2.7	Servomotor modelo MPL-A310F-SJ22AA.....	9
2.8	Sensor indutivo 871T-R8B18 [2]	10
2.9	Câmera da Banner Engineering utilizada no projeto [2]	10
2.10	Exemplo de programa <i>ladder</i> com instruções de movimentação.....	14
3.1	RSLinx com câmera reconhecida	17
3.2	Programa PresencePLUS para calibração da câmera	18
3.3	Referência de Posição para Excursão de 30cm	20
3.4	Referência de Velocidade para Excursão de 30cm	20
3.5	Resultado com Velocidade Modelada para Excursão de 30cm	20
3.6	Resultado com Velocidade Constante para Excursão de 30cm.....	20
3.7	Referência de Posição para Excursão de 20cm	21
3.8	Referência de Velocidade para Excursão de 20cm.....	21
3.9	Resultado com Velocidade Modelada para Excursão de 20cm	21
3.10	Resultado com Velocidade Constante para Excursão de 20cm	21
3.11	Malha fechada de controle.....	22
3.12	Resultado com Malha Fechada Proporcional e $K_p = 0.0025 \frac{u}{mm \cdot s}$	22
3.13	Resultado com Malha Fechada Proporcional e $K_p = 0.0050 \frac{u}{mm \cdot s}$	22
I.1	<i>Trigger</i> da câmera	35
I.2	Parada de emergência	35
I.3	Detecção da posição horizontal da bolinha.....	36

LISTA DE TABELAS

2.1	Principais instruções <i>ladder</i>	13
2.2	Principais instruções de controle de movimento em <i>ladder</i>	13
2.3	Dados para simulação em escala real [2]	15
2.4	Dados para simulação em escala laboratorial.....	15
3.1	Relações mm/px para diferentes seções da barra de alumínio	18
3.2	Dados de calibração do servomotor, média obtida é de 71.32 mm/unidade	19

LISTA DE SÍMBOLOS

Símbolos Latinos

v	Velocidade linear	[m/s]
r	Referência de posição	[m]

Símbolos Gregos

ω	Velocidade angular	[rad/s]
----------	--------------------	---------

Sobrescritos

.	Variação temporal
-	Valor médio

Siglas

PCI	<i>Peripheral Component Interconnect</i>
CPU	Unidade Central de Processamento - <i>Central Processing Unit</i>
CAN	<i>Control Area Networking</i>
CIP	Protocolo Industrial Comum - <i>Common Industrial Protocol</i>
CLP	Controlador Lógico Programável
PWM	Modulação por Largura de Pulso - <i>Pulse Width Modulation</i>
SI	Sistema Internacional de Unidades
EDS	<i>Electronic Data Sheet</i>
RSLogix	<i>Rockwell Software Logix 5000</i>
RSLogix5000	<i>Rockwell Software Logix 5000</i>

Capítulo 1

Introdução

A motivação do presente trabalho aparece na área petrolífera, nas operações de reentrada de risers e o objetivo é a validação de um sistema de controle em malha fechada em escala laboratorial como uma alternativa ao modelo manual empregado atualmente.

1.1 Contextualização

O petróleo tem importância econômica global. Uma das formas de extração do mesmo é aquela feita em águas profundas, a qual o Brasil tem feito importantes avanços desde a descoberta do Pré-Sal em 2006. Em abril de 2015, chegou-se à produção de mais de 800 mil barris por dia no pré-sal, com campos situados em águas profundas e ultraprofundas [3]. Os desafios nesta área da engenharia são enormes, pois as operações são muito complexas.

Na Figura 1.1, observa-se uma das operações necessárias para a extração no Pré-Sal, especificamente a operação de reentrada. Nesta operação, um *riser* deve ser conectado da plataforma até o poço de petróleo no leito oceânico. O comprimento do *riser* chega a 2km.

Devido à complexidade inerente das diversas operações *offshore*, propulsores e sensores de localização e orientação - GPS, giroscópios, câmeras, etc - são requisitos essenciais para se poder posicionar a embarcação e os *risers* [2].

1.2 Objetivos do projeto

O foco deste trabalho está na operação de reentrada, conforme apresentada na Figura 1.1. Atualmente, é uma operação feita manualmente por um operador na plataforma que observa remotamente as imagens capturas por *ROVs* (*Remotely Operated Vehicles*) da região do *riser* próxima ao poço e controla a plataforma através de um *joystick* com o auxílio do sistema de posicionamento dinâmico. O custo envolvido na operação é enorme e os riscos para o equipamento também, já que os próprios *ROVs* e o *riser* ficam sujeitos às perturbações das ondas, correntezas

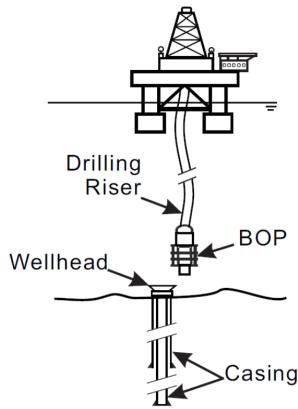


Figura 1.1: Operação de reentrada [1]

e variações ambientais no fundo do mar. A Figura 1.2 apresenta o esquema que acabou de ser descrito.

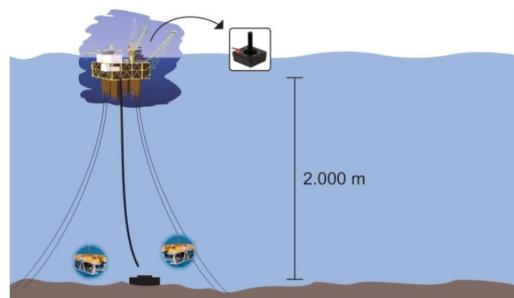


Figura 1.2: Método atual para reconexão no poço [2]

O objetivo deste trabalho é apresentar uma forma mais eficiente de realizar a operação de reentrada, economizando tempo e recursos, evitando riscos para pessoal e equipamento. Rédytton [2] validou o controle em malha aberta. Neste presente trabalho, deseja-se revalidar a malha aberta, com pequenas mudanças na excursão total do carrinho em comparação com o trabalho de Rédytton, assim como realizar as configurações necessárias para fechar a malha.

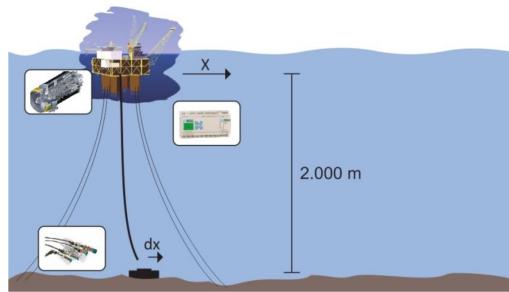


Figura 1.3: Método proposto para reconexão no poço [2]

Para fechar a malha, necessita-se de um sensor que realmente os dados de posição, o qual, na

presente bancada, é uma câmera industrial. Passos necessários incluem:

- Alimentação da câmera;
- Atualização do Firmware da Câmera
- Conexão na rede Ethernet/IP da câmera e CLP;
- Configuração da câmera no software RSLogix;
- Alimentação da bancada com fontes dedicadas de 24V¹

¹Anteriormente, duas fontes de tensão de saída ajustável eram utilizadas, mas foram trocadas por fontes de saída de tensão única.

Capítulo 2

Fundamentos

Este capítulo apresenta equações básicas do sistema que se deseja validar, assim como informações sobre a bancada laboratorial e também sobre a programação do CLP.

2.1 Equações Governantes

Estruturas submarinas tais como os *risers* são esbeltas e tem um alto módulo de cisalhamento. Portanto, a simplificação de Euler-Bernoulli para vigas é utilizada para propósitos de modelagem. O deslocamento de interesse é o horizontal e o *riser* está sob a ação de forças hidrodinâmicas externas e de tração. A equação diferencial parcial para a variável deslocamento, Υ , é dada por

$$m_s \frac{\partial^2 \Upsilon}{\partial t^2} = -EJ \frac{\partial^4 \Upsilon}{\partial z^4} + \frac{\partial}{\partial z} \left(T(z) \frac{\partial \Upsilon}{\partial z} \right) + F_n(z, t), \quad (2.1)$$

na qual m_s é a densidade linear do tubo, E é o módulo de Young e J é o segundo momento de inércia do *riser*. $T(z)$ descreve as forças de tração ao longo do comprimento do *riser*. $F_n(z, t)$ é a força resultante externa [4].

As únicas forças externas atuando no *riser* são hidrodinâmicas, exceto nas extremidades do topo e do fundo, nas quais forças de reação seguem condições de contorno. A equação de Morison descreve a força externa resultante:

$$F_n(z, t) = -m_f \frac{\partial^2 \Upsilon}{\partial t^2} - \mu \left| \frac{\partial \Upsilon}{\partial t} \right| \frac{\partial \Upsilon}{\partial t}, \quad (2.2)$$

na qual m_f é a massa do fluido adicionado e μ é o coeficiente de arrasto.

Para o presente trabalho, apenas o resultado do modelamento para malha aberta foi fornecido e utilizado. Para trabalhos futuros, as equações de controle serão desenvolvidas de acordo com as exigências do projeto.

2.2 Controle

Técnicas de controle simples devem ser introduzidas de forma a se compreender o objetivo deste trabalho, que é do posicionamento do *riser* por meio de controle em malha fechada. Primei-

ramente, apresenta-se o controle em malha aberta, cujo diagrama pode ser observado na Figura 2.1. A variável $r = r(t)$ é a referência do sistema. Neste tipo de controle, a saída não é realimentada na entrada. Desta forma, este tipo de controle não requer sensores, pois somente dá uma referência de entrada que a planta deve seguir. Caso haja erros para seguir a trajetória, eles não poderão ser compensados e é um controle mais recomendado quando o sistema é preciso e há pouca ou nenhuma perturbação. No entanto, este não é o caso do *riser*, pois o movimento das águas no leito oceânico perturba o tubo, causando erros na posição final desejada.

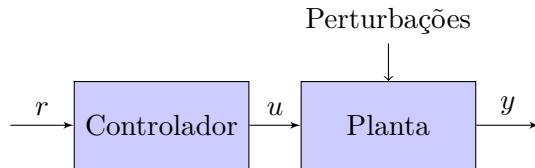


Figura 2.1: Malha aberta de controle

Uma forma de se compensar as perturbações do ambiente é realimentando a saída na entrada, calculando a diferença entre a referência e o valor medido. Assim, um valor de erro $e = e(t)$ é obtido e o sistema calcula o sinal u conforme o erro evolui. A Figura 2.2 mostra um esquema básico deste sistema.

O controle malha aberta foi anteriormente verificado na referência [2].

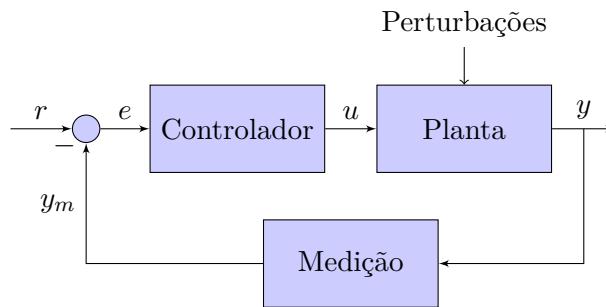


Figura 2.2: Malha fechada de controle

2.3 Bancada

A bancada da ponte rolante presente no Laboratório de Automação e Controle está esquematizada na Figura 2.3. Serão detalhados os componentes desta bancada, de modo a se entender o papel de cada um deles. Observe que no esquemático falta a câmera, que é um sensor que fica de frente para a bancada.

2.3.1 Controlador Lógico-Programável

O controlador lógico-programável (CLP - ver Figura 2.4) é a espinha dorsal da bancada. Ele é responsável por executar os comandos de controle sobre todos os elementos que estão conectados

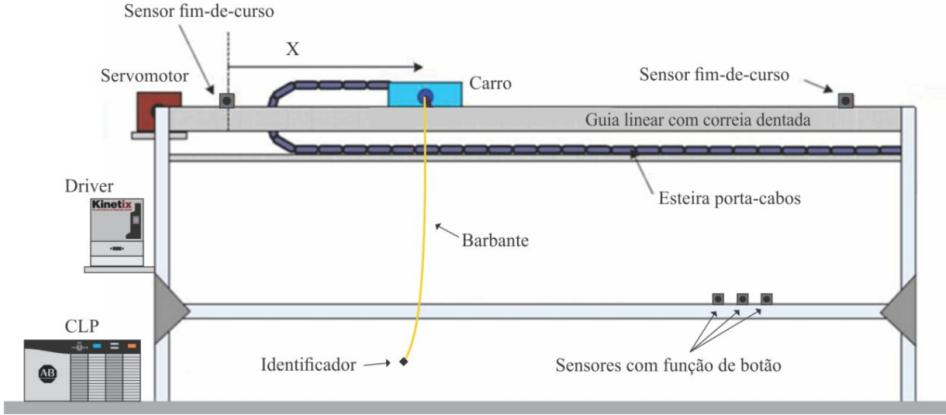


Figura 2.3: Esquemático da Bancada Utilizada para o Experimento [2]

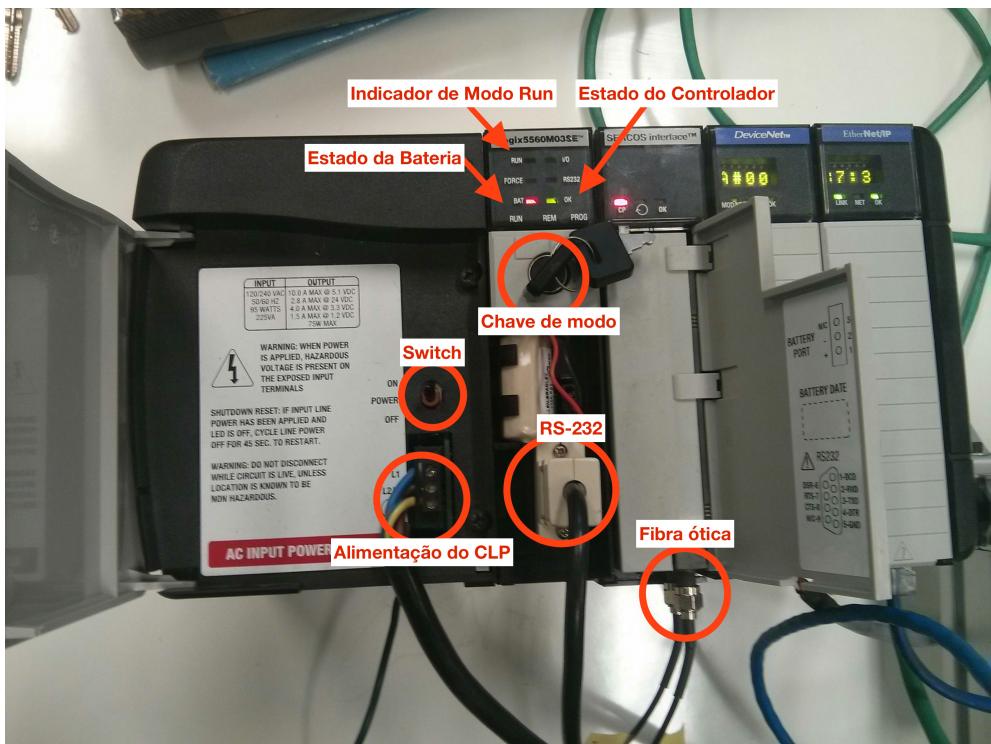


Figura 2.4: CLP com identificação de elementos

a ele. A programação do CLP é realizada via computador, mas, uma vez feito o *download* do programa ao CLP, a execução ocorre independentemente do computador, contanto que o CLP esteja em modo de execução.

O CLP utilizado é fabricado pela *Allen Bradley*, modelo Logix5560M03SE. Tal modelo possui memória lógica e de dados de 750 KiB, e memória de *I/O* de 494 KiB. Há quatro módulos no *chassis* do controlador:

- O próprio controlador;
- *SERCOS Interface*;

- DeviceNET;
- EtherNet/IP.

Cada um desses módulos será apresentado posteriormente com mais detalhes. Além dos módulos, o controlador ainda possui um *switch* liga/desliga presente no *chassis*. No módulo Logix, há uma chave responsável por alterar o modo de funcionamento do mesmo. As posições possíveis dessa chave são:

- RUN;
- REM;
- PROG;

Na prática, o modo REM se divide em dois modos: REM RUN e REM PROG. A maneira de se diferenciar os dois é observar, no módulo Logix, o estado do LED indicador de modo RUN quando a chave estiver na posição REM.

No modo RUN, o controlador apenas roda o programa presente em sua memória; não há qualquer comunicação remota. No modo PROG, o controlador não roda nenhum programa; ele apenas pode receber um novo código. Nos modos REM, há a comunicação com o computador, permitindo verificar valores de variáveis de interesse e alterar, se necessário, o programa a ser rodado pelo controlador. O programa presente no controlador, em modo REM, só roda o código se estiver no modo REM RUN; se for necessário atualizar o programa, o modo deve ser o REM PROG.

2.3.2 SERCOS Interface

Sercos é um barramento digital de automação que interconecta controladores, *drives*, dispositivos de entrada/saída e atuadores para máquinas e sistemas controlados numericamente. Foi projetado para comunicação serial de alta velocidade de dados em sistemas de tempo real por meio de fibra ótica (Sercos I & II) ou um cabo Ethernet Industrial (Sercos III). Sercos é um padrão internacional [5].

Num sistema Sercos, todas as malhas que contém servomotores são normalmente fechadas no *drive*. Isto reduz a carga computacional no CLP, permitindo-o sincronizar mais eixos do que conseguiria caso contrário. Além disso, fechar a malha dos servomotores com o *driver* ajuda a reduzir o efeito do atraso de transporte entre o controle de movimento e o *driver* [5].

Nesta bancada, o CLP deve se comunicar com o servomotor (MPL-A310F-SJ22AA) através do *drive* Kinetix (2094-AC05MP5), de forma a movimentar o carrinho segundo uma trajetória planejada ou segundo uma lei de controle em malha fechada executando no CLP. Essa comunicação se dá por um par de fibras óticas full-duplex, conforme se observa na Figura 2.4, o que caracteriza uma rede Sercos I ou II.

2.3.3 Line Interface Module 2094-AL09

Este módulo não apareceu no esquemático da Figura 2.3, mas ele tem a função essencial de interfacear a rede trifásica com o servo *drive*, permitindo o acionamento do motor. Na Figura 2.5, se observa que há três conjuntos de disjuntores nesse módulo: CB1 – liga ou desliga a rede trifásica do *drive* –, CB2 – fornece tensão monofásica ao servo *drive* – e CB3 – liga as duas fontes de tensão DC de 24V –, responsáveis pelas entradas e saídas digitais do módulo e alimentação do freio do motor [2].

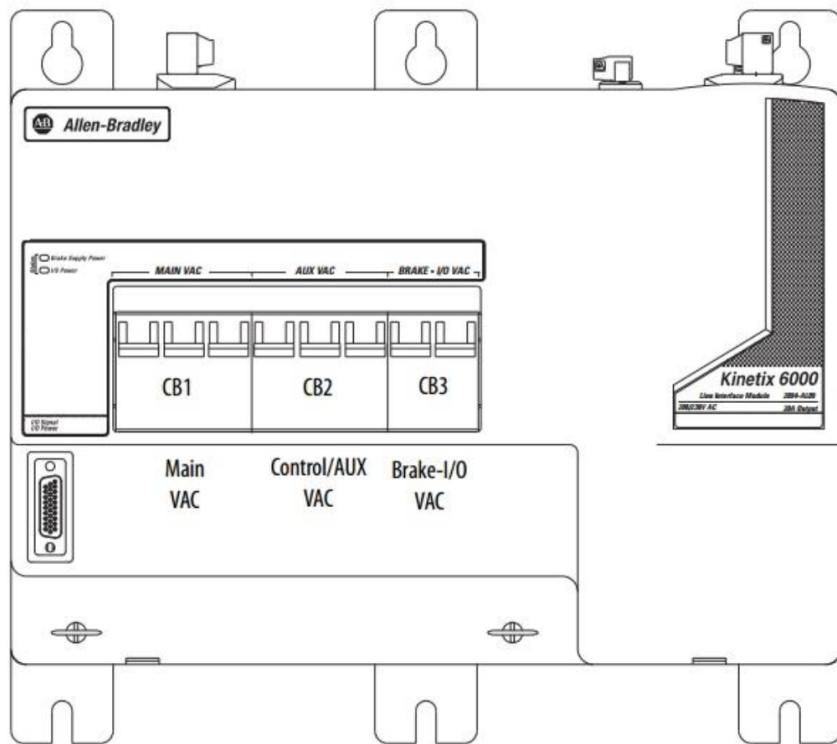


Figura 2.5: Line Interface Module modelo 2094-AL09 da Allen Bradley [2]

2.3.4 Drive Kinetix 6000 da Allen Bradley

Por meio da Interface Sercos, o CLP comanda o drive que então fornece potência ao motor. O drive controla o servomotor por meio de pulsos PWM. A Figura 2.6 apresenta um Drive Kinetix 6000 da Allen Bradley similar ao utilizado no laboratório. Mais detalhes sobre o funcionamento do drive estão disponíveis em [2] e [6].

2.3.5 Servomotor

Um servomotor é um atuador rotatório que permite controle preciso da posição angular. O motor consiste de um motor acoplado a um sensor para a realimentação de posição/velocidade. Também é necessário um drive servo para completar o sistema. O *drive* usa o sensor de realimentação para controlar precisamente a posição angular do motor, ou seja, é uma operação em



Figura 2.6: Drive Kinetix 6000 da Allen Bradley

malha fechada. Assim, usando servomotores em malha fechada, tem-se uma alternativa de alto desempenho aos motores de passo e de indução [7].

O servomotor MPL-A310F-SJ22AA da *Allen-Bradley* foi utilizado e está representado na Figura 2.7. O mesmo é composto por um motor indutivo de tensão nominal $230V_{ac}$ e um encoder do tipo StegmanHiperface, que mede posição de forma absoluta e velocidade de forma incremental [2].



Figura 2.7: Servomotor modelo MPL-A310F-SJ22AA

2.3.6 Sensores indutivos

Os sensores indutivos são elementos detectores de presença, particularmente de objetos metálicos. Eles funcionam através da variação de campo magnético ocasionada pela presença do objeto a ser identificado. Tal variação de campo magnético provoca uma variação de corrente dentro do sensor, alterando seu estado.

Na presente bancada, há 6 sensores indutivos da família 871TM, similares ao da Figura 2.8, fabricados pela *Allen-Bradley*. Eles são alimentados com tensão de 24 V, que está dentro dos

limites padrão. São sensores feitos de aço, adaptados a ambientes industriais.



Figura 2.8: Sensor indutivo 871T-R8B18 [2]

A importância desses sensores é enorme. O motivo é que a câmera às vezes falha para obter a posição atual e tem um limite de frequência devido ao processamento interno que ela tem de realizar. Com os sensores indutivos, tem-se um processamento rápido para identificar se o motor está na região próxima dos limites. Assim, a rotina de segurança que trava o motor depende diretamente desses sensores indutivos, como mostra o Anexo I.2.2.

2.3.7 Câmera PresencePlus

A câmera PresencePlus P4 GEO da Banner Engineering foi utilizada no projeto - veja Figura 2.9. Esta câmera é um sensor robusto utilizado em ambientes industriais com fácil utilização. Sua programação é feita em software próprio da Banner e é visual.

Dentre as capacidades da câmera, nota-se que ela pode capturar até 24 imagens por segundo. No entanto, além da captura das imagens há o processamento das imagens, que pode ser feito na própria câmera ou em um módulo externo da própria Banner, que também consome certo tempo. O programa utilizado no projeto está disponível no Anexo I.3.1.



Figura 2.9: Câmera da Banner Engineering utilizada no projeto [2]

2.4 Redes Utilizadas

2.4.1 DeviceNET

DeviceNET é um protocolo de baixo nível da camada de aplicação, voltado a ambientes industriais [8]. É responsável pela interconexão de dispositivos visando ao compartilhamento de dados. Foi desenvolvido pela *Allen-Bradley*, em cima da tecnologia CAN (*Control Area Networking*), desenvolvida pela *Bosch*. Tal rede suporta comunicação entre dispositivos de baixo nível, como sensores e atuadores, e dispositivos de alto nível, como o computador e o CLP [9].

O DeviceNET é uma combinação entre a camada física disponibilizada pelo CAN e um protocolo industrial, o CIP (*Common Industrial Protocol*), que rege as redes industriais em geral. Permite uma rápida configuração entre dispositivos a *byte*, suportanto tanto dispositivos analógicos quanto digitais [2]. Permite velocidades de transmissão de até 500 kbps, sendo bem mais lenta que uma rede Ethernet.

No experimento, a rede DeviceNET é utilizada para a conexão entre o CLP e os sensores indutivos presentes na planta [2]. Devido à flexibilidade dos sensores e da rede, suas informações podem ser trabalhadas tanto no modo analógico quanto no modo digital. Para fins de detecção de fim-de-curso, entretanto, o modo utilizado é o digital, uma vez que não será preciso determinar a distância entre sensor e carrinho; apenas é necessário verificar se o carrinho está na área de detecção do sensor. O módulo responsável pelo gerenciamento da rede DeviceNET é o 1756-DNB.

2.4.2 Ethernet/IP

A rede Ethernet/IP, introduzida em meados de 2001 pela ODVA (*Open DeviceNet Vendor Association*) [10], é um tipo de rede Ethernet voltada ao ambiente industrial, seguindo o CIP, assim como o DeviceNET [11]. É uma rede robusta, organizada segundo o modelo OSI de 7 camadas, que permite conexão com dispositivos conectados a redes Ethernet padrão; permite a passagem de dados via pacotes TCP ou UDP; é indicada em aplicações que exigem uma transferência rápida e confiável de dados, principalmente em aplicações de tempo real) [12].

O uso da rede Ethernet/IP é vantajoso no sentido de que permite a conexão entre vários nós ligados entre si [10], o que não é possível com o RS-232, por exemplo. Porém, devido ao próprio funcionamento da rede, fica mais difícil obter os dados, pois, ao contrário de uma entrada serial, é necessário lidar com toda uma estrutura baseada no TCP/IP, por exemplo. Além disso, como o Ethernet exige um tamanho mínimo de *frame* para transmissão de dados de cerca de 64 *bytes*, a eficiência da transmissão pode ser afetada.

No presente experimento, a rede Ethernet/IP é utilizada para receber dados de inspeção da câmera, notadamente a posição horizontal da bola de isopor presa ao barbante. Além disso, ela é utilizada na transferência de programas entre o computador e o CLP (através do módulo Ethernet 1756-ENBT/A), uma vez que ela provê uma comunicação mais rápida do que o RS-232.

2.5 Programação do CLP

2.5.1 Visão geral

Programação, em termos gerais, é aplicada na resolução de problemas. Em particular, a programação de CLPs busca resolver, no âmbito industrial, problemas relacionados à automação de processos. Essa resolução de problemas segue uma metodologia, de forma a se direcionar o projeto. Antes de se proceder à programação, faz-se necessário seguir os seguintes passos [13]:

1. Descrição do problema;
2. Detalhamentos e melhoria do processo;
3. Especificação dos atuadores e sensores da planta;
4. Elaboração do algoritmo;
5. Representação gráfica do algoritmo, quando aplicável;
6. Esquema funcional, quando aplicável;
7. Seleção dos módulos do controlador;
8. Programação, utilizando linguagens suportadas.

Ao se programar o CLP, deve-se ter em conta que o controlador executa sempre três ciclos [13]:

1. *Scan* de Entrada: Ciclo em que o controlador recebe todos os dados de seus módulos de entrada;
2. *Scan* do Programa: Ciclo em que o controlador processa as entradas recebidas e gera as saídas;
3. *Scan* de Saída: Ciclo em que o controlador envia os dados de saída para os módulos de saída.

Para o presente experimento, entre as várias linguagens disponíveis para CLPs, foram selecionadas duas: uma linguagem gráfica, o ***ladder***, e uma linguagem textual, o **texto estruturado**. Ambas foram selecionadas por serem linguagens muito utilizadas, normalmente rápidas e que ocupam pouca memória, ao contrário de linguagens como SFC (*Sequential Flow Chart*) e FBD (*Function Block Diagram*), que também poderiam ser utilizadas.

2.5.2 Linguagem *ladder*

2.5.2.1 Introdução ao *ladder*

A linguagem *ladder* é uma linguagem de programação gráfica, e uma das primeiras a ser utilizada na programação de CLPs. Seu nome vem do inglês *ladder*, que significa escada; nome

dado em razão dos programas, ao serem feitos, assumirem a forma de uma escada, e lidos de cima para baixo (movimento de descida). Essa linguagem foi estruturada de forma a ter uma simbologia semelhante à de um diagrama de conexão de relés, que eram utilizados em indústrias antes do CLP.

Todo programa *ladder* possui duas linhas verticais e, entre elas, uma ou mais linhas horizontais nas quais são especificados os comportamentos do programa. A linguagem possui várias instruções, em sua maioria comuns entre diferentes tipos de CLP. A Tabela 2.1 mostra as principais instruções utilizadas em *ladder*, utilizando a sintaxe presente no *software* RSLogix5000.

Tabela 2.1: Principais instruções *ladder*

Desenho da Instrução	Nome da Instrução	Descrição
- () -	OTE	Atualiza variável booleana de acordo com a condição da linha
- (L) -	OTL	Atribui valor verdadeiro à variável booleana se a linha for alimentada
- (U) -	OTU	Atribui valor falso à variável booleana se a linha for alimentada
- [] -	XIC	Examina se a variável possui valor verdadeiro
- [/] -	XIO	Examina se a variável possui valor falso

2.5.2.2 Instruções específicas

No presente trabalho, a presença do servomotor exige uma lógica e entradas que não são booleanas, ou seja, que não podem ser tratadas pelos operadores descritos na Tabela 2.1. Portanto, faz-se necessário o uso de novas instruções, diretamente relacionadas com o movimento do atuador. Tais instruções são conhecidas, dentro do RSLogix5000, como *Motion Control Instructions*, ou instruções de controle de movimento. A Tabela 2.2 mostra as instruções desse tipo utilizadas no experimento.

Tabela 2.2: Principais instruções de controle de movimento em *ladder*

Nome da Instrução	Sigla	Função
<i>Motion Servo On</i>	MSO	Inicializa o servomotor, travando a esteira.
<i>Motion Axis Jog</i>	MAJ	Envia comandos de velocidade e sentido de rotação ao motor, executando sua movimentação.
<i>Motion Axis Stop</i>	MAS	Interrompe a movimentação do motor; a esteira permanece travada.
<i>Motion Servo Off</i>	MSF	Desativa o servomotor, destravando a esteira e permitindo a movimentação manual do carrinho.

As funções descritas pela Tabela 2.2 lidam com outros tipos de variáveis, como números reais, para a velocidade do motor, e até mesmo estruturas que o *software* RSLogix cria para armazenar

informações sobre os eixos e o controle feito pelas instruções. Tais parâmetros podem ser alterados dentro dos blocos das funções. A Figura 2.10 mostra um exemplo de tais instruções.

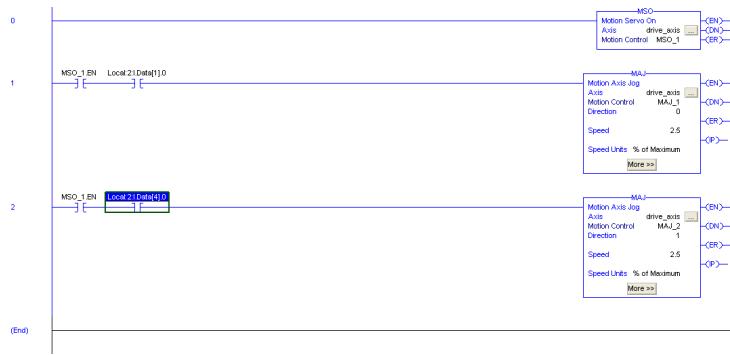


Figura 2.10: Exemplo de programa *ladder* com instruções de movimentação.

2.5.3 Texto Estruturado

2.5.3.1 Visão geral do texto estruturado

Além de *ladder*, que é uma linguagem gráfica, o CLP utilizado no experimento também suporta uma linguagem puramente textual, o texto estruturado. Essa linguagem é uma linguagem sequencial, que lembra muito linguagens como BASIC, C ou Pascal. Cada comando é executado em sequência, salvo em caso de desvios (estruturas do tipo *if-then-else*) ou repetições (*loops*, estruturas do tipo *while*).

Em relação ao *ladder*, o texto estruturado possui vantagens e desvantagens. A principal vantagem é o fato do texto estruturado se assemelhar às linguagens de programação mais utilizadas; isso torna tal linguagem de fácil aprendizado. Além disso, por ser texto, caso seja necessário editar muitos parâmetros cuja criação seja de difícil automação, um programa auxiliar escrito em linguagens de programação genéricas pode simplificar a geração do texto estruturado. Porém, uma desvantagem do texto em relação ao *ladder* reside nas funções de movimentação, uma vez que, ao contrário do *ladder*, o texto estruturado não dá uma indicação clara associando um valor e a variável que ele representa, além da ordem dos parâmetros. Isso torna o uso de funções como MAJ (vide Tabela 2.2) um pouco mais complicado em texto estruturado.

2.5.3.2 Instruções de movimentação do motor para o texto estruturado

Assim como o *ladder*, o texto estruturado permite o uso das funções descritas na tabela 2.2. A sintaxe de uso dessas funções lembra muito a sintaxe da linguagem C, com o nome da função e uma lista de parâmetros válidos separados por vírgulas. A seguir, um exemplo de uso:

```
/**
 * MSO: ativa o servo cujo eixo eh descrito
 * por drive_axis; informacoes de controle
 * sao gravadas em MSO_1
```

```

*/  

MSO( drive_axis ,MSO_1 );  

/* Atribui o valor 0.0 ao primeiro elemento do array speed */  

speed [0] := 0.0;  

/* Atribui 1 para dataInitialized */  

dataInitialized := 1;

```

Nota-se que o código apresentado, a menos da operação de atribuição, possui uma sintaxe muito semelhante à de C. A função MSO acima recebe dois parâmetros, assim como em *ladder*; porém, a única indicação de como os parâmetros se comportam é a ordem com que eles são colocados; no caso de MSO, o 1º parâmetro é o eixo que será inicializado, e o 2º parâmetro é uma *tag* que guarda as informações de controle executadas por MSO.

2.6 Parâmetros para gerar trajetórias

Fabrício et al [4] desenvolveram um método para gerar trajetórias *offline* (controle malha aberta) assim como *online* (controle malha fechada). Para se utilizar do método, é necessário ter parâmetros do sistema. Se se fosse simular ou controlar o sistema real, utilizariam-se os dados da Tabela 2.3, conforme referência [2].

Tabela 2.3: Dados para simulação em escala real [2]

Dados do Riser	
Diâmetro externo	0.55m
Diâmetro interno	0.5m
Comprimento	2000m
Módulo de Elasticidade	200GPa
Densidade	7860kg/m ³
Dados do fluido (Água)	
Densidade do fluido	1000kg/m ³
Viscosidade dinâmica	$10^3 \text{ Pa} \cdot \text{s}$

Tabela 2.4: Dados para simulação em escala laboratorial

Dados da Massa na Ponta - Isopor	
Diâmetro Externo	30.6mm
Densidade	10kg/m ³
Dados do Riser (Barbante)	
Diâmetro externo	2mm
Diâmetro interno	0mm
Comprimento	82cm
Módulo de Elasticidade	2.1MPa
Densidade	191kg/m ³
Dados do fluido (Ar)	
Densidade do fluido	1.2754kg/m ³
Viscosidade dinâmica	$17.2 \cdot 10^6 \text{ Pa} \cdot \text{s}$

Como é importante se validar o sistema, mas nem sempre isso é possível em escala real devido ao altíssimo custo e aos riscos envolvidos, é necessário um conjunto equivalente de parâmetros tais que permita validar o sistema em escala laboratorial. O *riser* é representado por um barbante e a água é trocada pelo ar. Uma massa na ponta foi adicionada e os dados resultantes estão na Tabela 2.4.

Capítulo 3

Resultados

Este capítulo apresenta os procedimentos experimentais realizados e seus resultados.

3.1 Câmera

Os problemas com a câmera se resumiram a configuração da rede, calibração e programação, sendo o mais difícil esse último.

3.1.1 Configuração da Rede

A câmera estava com um *firmware* antigo e foi necessário obter o arquivo 2014R1B PresencePLUS Firmware¹ que é o programa de atualização da *Banner Engineering* para esta câmera. Bastou executá-lo no computador, estando a câmera conectada ao *switch* do laboratório assim como o computador estava, ambos por cabos Ethernet. Antes desta atualização, o *software* da câmera não permitia selecionar a opção Ethernet/IP de forma a permitir utilizar o módulo Ethernet/IP do CLP.

Uma vez atualizado o *firmware*, foi obtido o arquivo EDS² da câmera que foi então integrado ao *software* da Rockwell no computador por meio do programa *EDS Hardware Installation Tool* da própria Rockwell, parte do RSLinx.

Após os procedimentos anteriores, adiciona-se a câmera como um módulo genérico usando o *software* RSLogix, conforme instruções da Banner Engineering [14]. Dentro das configurações realizadas no RSLogix Para se efetuar a configuração da câmera, é necessário utilizar o RSLinx para verificar se a mesma está conectada ao computador e o *software* da PresencePlus, para identificar a câmera pelo endereço IP, uma vez que a rede utilizada é Ethernet/IP. A Figura 3.1 mostra a janela do RSLinx com a câmera reconhecida.

¹ Atualização de Firmware da Câmera - http://info.bannerengineering.com/_dav/cs/idcplg?IdcService=GET_FILE&RevisionSelectionMethod=LatestReleased&dDocName=B_4170042. Acesso em 29/11/2015.

² Arquivo EDS disponível em <http://www.bannerengineering.com/en-US/products/sub/78#ui-tabs-37>. Acesso em 29/11/2015.

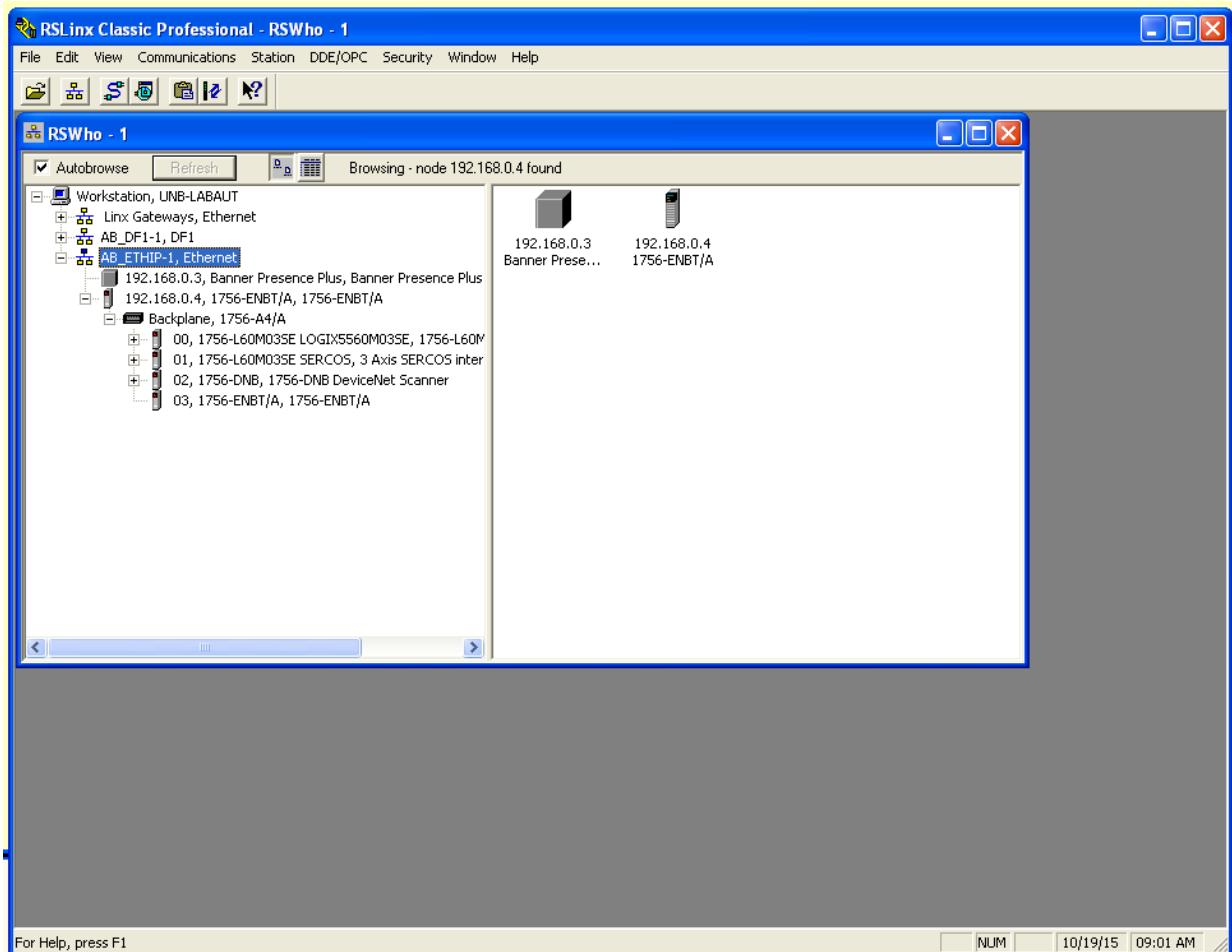


Figura 3.1: RSLinx com câmera reconhecida

3.1.2 Calibração

A câmera permite fazer medidas de distâncias em *pixels*. De forma a se converter essa distância para milímetros, uma barra de alumínio com marcas e tamanho conhecido é utilizada. É importante primeiro calibrar o sistema, para se saber se há deformação de pixels significante ao longo da distância de interesse, nomeadamente o tamanho da barra de alumínio sendo utilizado, cerca de 532mm. No PresencePlus P4 GEO 1.3, um programa é feito, com imagem de referência conforme Figura 3.2, que usa várias ferramentas de detecção de borda (ferramenta **Locate**) para identificar as posições de cada uma das marcas pretas da barra. Seis marcas foram feitas e a Tabela 3.1 apresenta os resultados para cada seção. A distância entre duas marcas é de 10cm, com exceção da distância entre P0 e PEND que é o comprimento total da barra.

O maior desvio da quantidade de milímetros por *pixel* das seções em relação à da barra inteira é de aproximadamente 4.93%. Há algumas imprecisões na maneira como os traços foram desenhados e é possível que o erro seja menor.

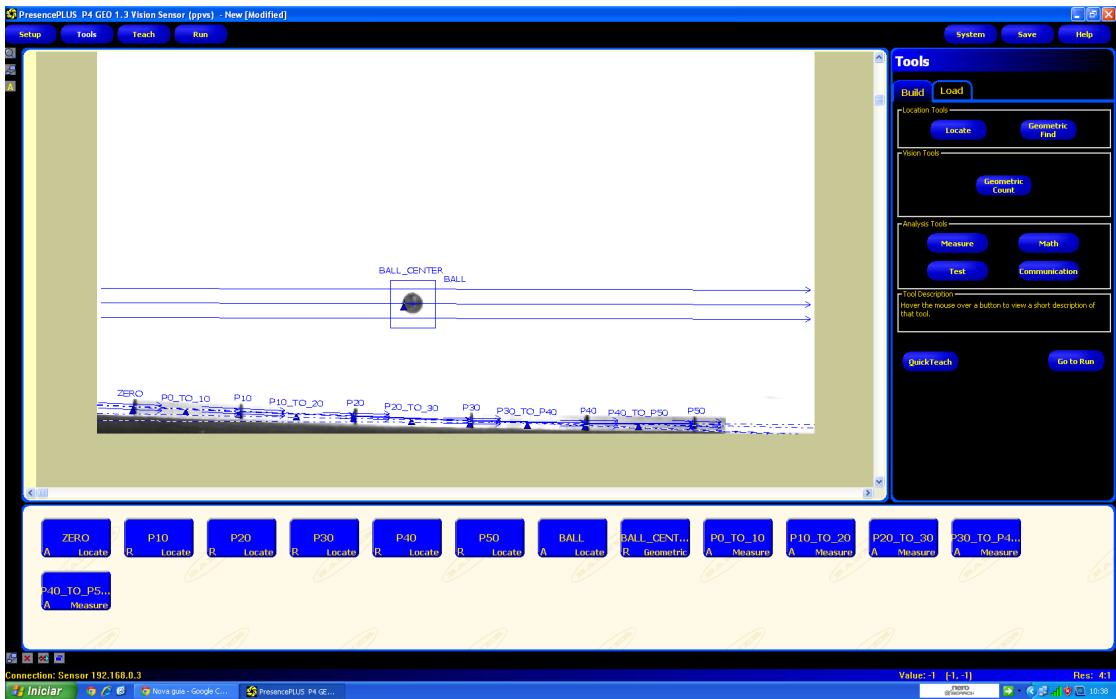


Figura 3.2: Programa PresencePLUS para calibração da câmera

Tabela 3.1: Relações mm/px para diferentes seções da barra de alumínio

Seção 1	Seção 2	Distância (px)	mm/px
P0	P10	160	0.625
P10	P20	173	0.578
P20	P30	176	0.568
P30	P40	173	0.578
P40	P50	163	0.613
P0	PEND	893	0.596

3.1.3 Programação

A programação da câmera inicia obtendo uma imagem de referência e adicionando-se ferramentas de detecção de pontos de interesse, como mostra a Figura 3.2. A ferramenta **Locate** é responsável por detectar as bordas da bolinha, e a ferramenta **Geometric** detecta o centro da mesma. Também é possível adicionar ferramentas que fazem operações matemáticas (ferramenta **Math**), executam medições (ferramenta **Measure**), assim como as que enviam dados pela rede (ferramenta **Communication**), o que é essencial para comunicar com o CLP.

3.2 Calibração do Servomotor

O RSLogix tem o bloco **MAJ – Motion Axis Jog** – que permite alterar a velocidade do motor enquanto ele se movimenta. No entanto, o bloco espera que a entrada seja do tipo [u/s] ao invés de

alguma unidade no SI tal como [mm/s]. Devido a isso, foi necessária uma calibração do sistema. Nela, anotou-se a posição inicial x_0 e a posição final x_f , ambas em milímetros, e definia-se um tempo Δt no qual o carrinho se movimentaria a uma velocidade v em [u/s]. Daí, calculava-se a velocidade em [mm/s] com esses dados e tirou a média de alguns ensaios para se obter o valor de uma unidade, que é aproximadamente 71.32mm. Os dados de calibração estão na Tabela 3.2.

Tabela 3.2: Dados de calibração do servomotor, média obtida é de 71.32 mm/unidade

x_0 - [mm]	x_f - [mm]	Δt - [s]	Velocidade - [u/s]	Velocidade - [mm/s]	mm/u
2	71.8	2	0.5	34.9	69.8
6	76.1	2	0.5	35.05	70.1
6	188	5	0.5	36.4	72.8
6	185	2.5	1	71.6	71.6
6	77	10	0.1	7.1	71
6	296.5	20	0.2	14.525	72.625

3.3 Malha aberta

O planejamento de trajetória em malha aberta e malha fechada para o *riser* que está sendo utilizado neste trabalho foi desenvolvido por Fabrício et al [4], conforme mencionado anteriormente. Um programa em MATLAB foi escrito e gerou trajetórias de excursão pré-definida. Rédytton [2] testou o sistema para uma excursão de cerca de 1m, que é maior que o tamanho do barbante (82cm) conforme apresentado na Tabela 2.4. No entanto, seu trabalho não utilizou uma massa de isopor na ponta, daí existirá uma diferença entre os resultados. Observe que, em malha aberta, deixou-se o ar condicionado da sala desligado, pois isto seria uma perturbação.

3.3.1 Excursão de 30cm

O primeiro teste em malha aberta testou a trajetória de posição da Figura 3.3. Conforme Rédytton [2] mencionou em seu trabalho, utilizar os módulos de posição é mais lento do que utilizar módulos de velocidade para seguir essa trajetória. Desta forma, optou-se por usar diferenças finitas para derivar essa trajetória, resultando na Figura 3.4.

Com um período de cerca de 41ms, uma tarefa eventual executava no CLP, aplicando um certo valor de velocidade ao carrinho. A câmera leu os valores de posição cada vez que a tarefa iniciava e o software RSLogix desenhou os dados, que podem ser observados na Figura 3.5. Observa-se que a posição evolui de forma bem suave. Para se realizar uma comparação, calculou-se o tempo que o carrinho se move na trajetória (Δt , tempo no qual a velocidade não é nula) e calculou-se $\bar{v} = \frac{\Delta x}{\Delta t}$. No caso presente, $\Delta x = 30\text{cm}$ e $\Delta t \approx 3\text{s}$ (veja Figura 3.4), resultando em $\bar{v} = 100\text{mm/s}$. A Figura 3.6 apresenta esses resultados e observa-se várias oscilações quando o carrinho para, diferente do caso anterior.

Vídeos foram criados para cada um destes casos e estão disponíveis no YouTube®®, tanto para

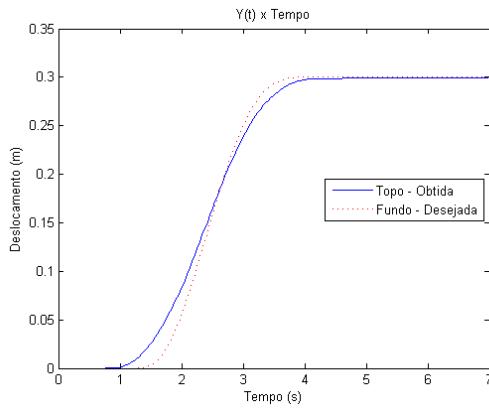


Figura 3.3: Referência de Posição para Ex- cursão de 30cm

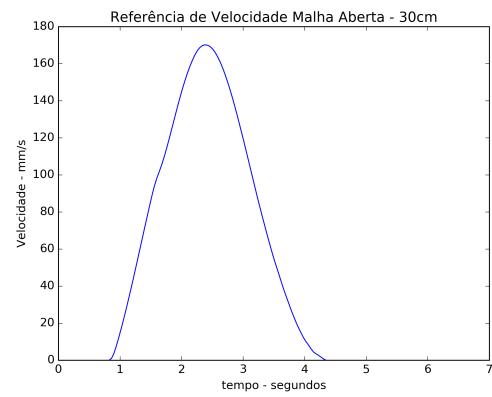


Figura 3.4: Referência de Velocidade para Excursão de 30cm

a trajetória modelada³ quanto para a trajetória com velocidade constante⁴.

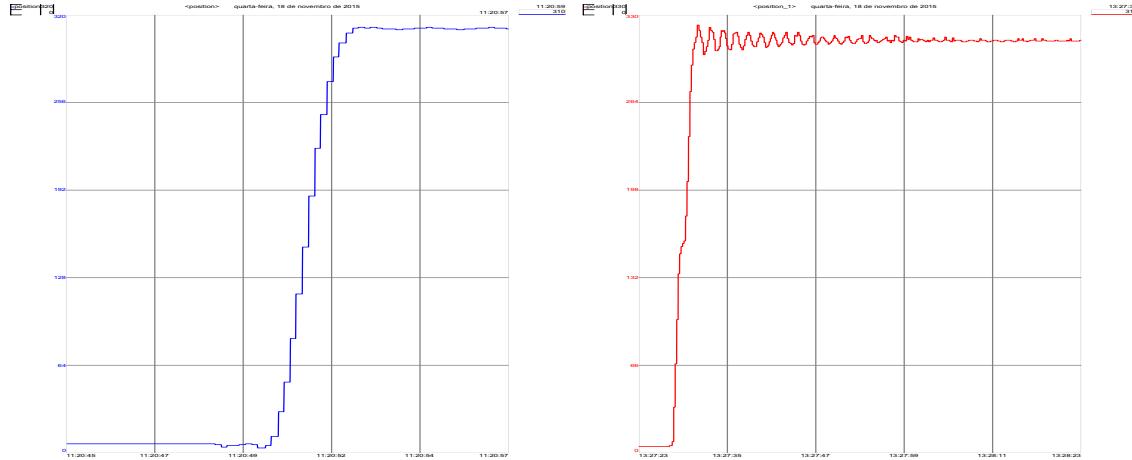


Figura 3.5: Resultado com Velocidade Mo- delada para Excursão de 30cm

Figura 3.6: Resultado com Velocidade Cons- tante para Excursão de 30cm

3.3.2 Excursão de 20cm

Na prática, o diâmetro do *riser* e de sua massa é bem menor que o comprimento do mesmo. Daí, é importante testar uma menor excursão. Além disso, esse teste terá menor tempo de movimentação e resultará numa velocidade média maior que o caso anterior. Desta forma, maiores oscilações são esperadas e o controle se torna mais difícil.

A trajetória de posição obtida é apresentada na Figura 3.7 e a sua derivada é apresentada na Figura 3.8. O período da tarefa eventual foi escolhido em 100ms. O resultado experimental⁵

³Controle Malha Aberta Modelado, 30cm - <https://youtu.be/1Kajz6LyauE>. Acesso em 29/11/2015.

⁴Controle Malha Aberta a Velocidade Constante, 30cm - <https://youtu.be/tBOTsmBcfVg>. Acesso em 29/11/2015.

⁵Controle Malha Aberta Modelado, 20cm - https://youtu.be/wg1Wq_6VRSG. Acesso em 29/11/2015.

apresentou mais oscilações que o caso anterior, mas está bem controlado, conforme se vê na Figura 3.9. O resultado com velocidade constante⁶ está na Figura 3.10 e nota-se que o resultado ficou bem pior. A velocidade média foi cerca de $\bar{v} \approx \frac{200\text{mm}}{1.5\text{s}} = 133.3\text{mm/s}$.

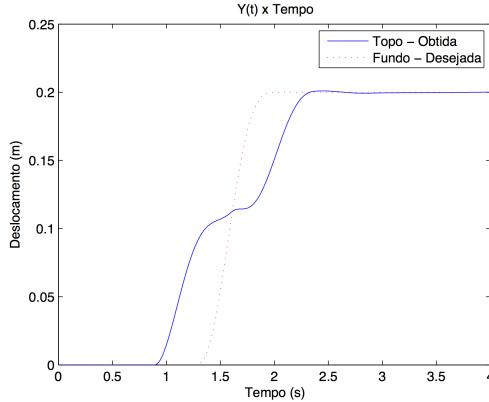


Figura 3.7: Referência de Posição para Exercício de 20cm

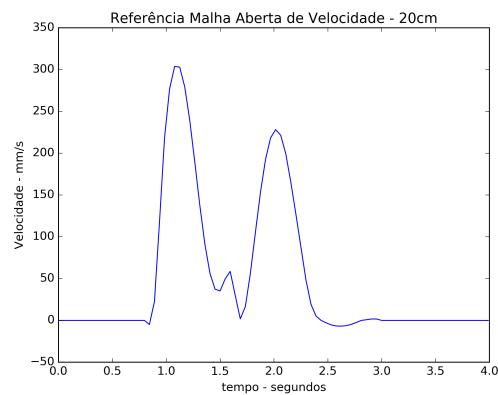


Figura 3.8: Referência de Velocidade para Excursão de 20cm

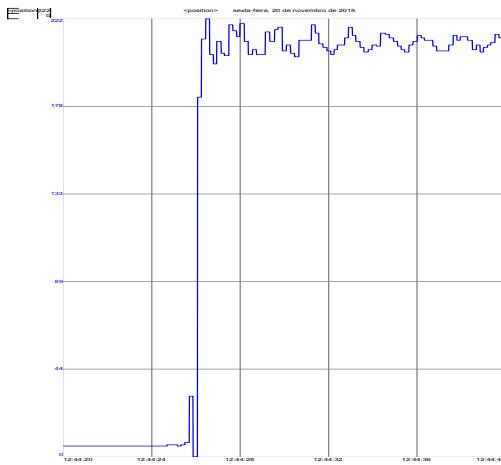


Figura 3.9: Resultado com Velocidade Modulada para Excursão de 20cm

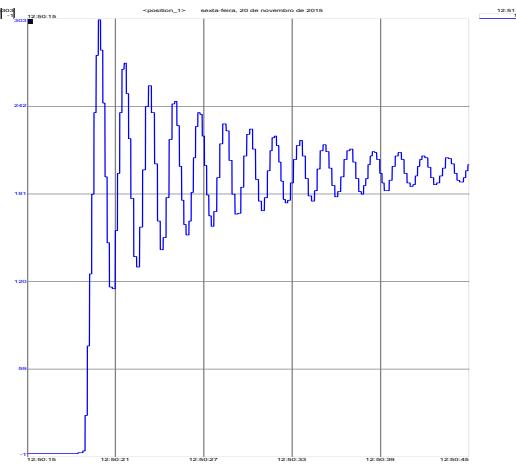


Figura 3.10: Resultado com Velocidade Constante para Excursão de 20cm

3.4 Malha fechada

O controle em malha fechada deve ser capaz de compensar por perturbações, pois o sensor identifica o resultado da atuação e realimenta a informação no sistema. Neste trabalho, conseguiu-se configurar a câmera adequadamente e faz-se um teste em malha fechada de um controlador bem simples: um controlador proporcional.

⁶Controle Malha Aberta a Velocidade Constante, 20cm - <https://youtu.be/Ges2-eYy69k>. Acesso em 29/11/2015.

3.4.1 Controlador P

O controlador proporcional utilizado tem o esquema conforme Figura 3.11. A referência de posição $r = r(t)$ é a entrada do sistema e a saída medida pela câmera, y_m , é subtraída da referência para resultar no erro e , cuja unidade é dada em milímetros. A entrada da planta, v , é dada em [u/s], conforme discutido anteriormente na subseção 3.2. Desta forma, a unidade de K_p é [$\frac{u}{mm \cdot s}$].

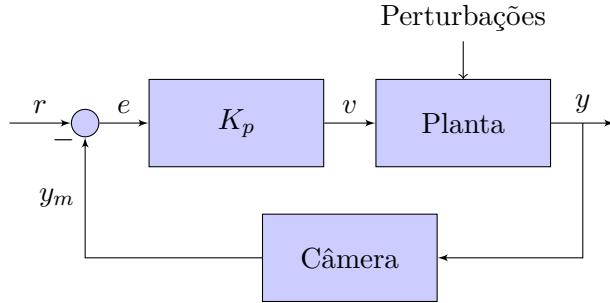


Figura 3.11: Malha fechada de controle

Valores de K_p foram escolhidos empiricamente e o resultado para $K_p = 0.0025$ ⁷ está na Figura 3.12 enquanto para $K_p = 0.0050$ ⁸ está na Figura 3.13.

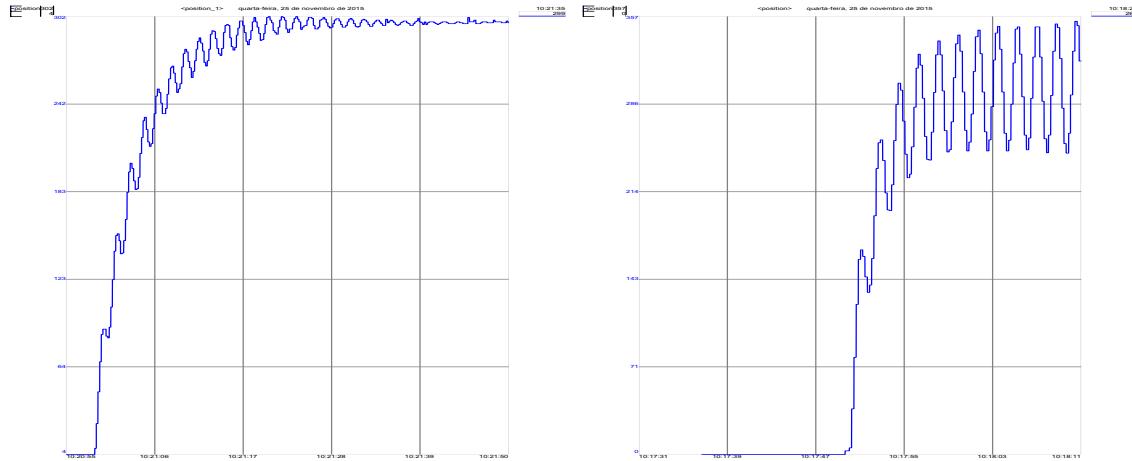


Figura 3.12: Resultado com Malha Fechada Proporcional e $K_p = 0.0025 \frac{u}{mm \cdot s}$

Figura 3.13: Resultado com Malha Fechada Proporcional e $K_p = 0.0050 \frac{u}{mm \cdot s}$

O período de amostragem utilizado foi de 100ms e o ar condicionado estava ligado nas duas situações apresentadas e nota-se que o primeiro resultado é bem melhor que o segundo, tendo menos oscilações. No entanto, este resultado está aquém do obtido em malha aberta.

⁷Controle Malha Fechada com $K_p = 0.0025$ - <https://youtu.be/mmT1ZwFBJ4s>. Acesso em 29/11/2015.

⁸Controle Malha Fechada com $K_p = 0.0050$ - <https://youtu.be/sHh3yvBrek4>. Acesso em 29/11/2015.

Capítulo 4

Conclusões

No início do trabalho, não havia nenhum conhecimento prévio dos autores sobre programação em tempo real utilizando CLPs, nem sobre programação CLP em geral. Nenhum conhecimento prévio da bancada existia também. No estágio atual, pode-se dizer que há relevante facilidade em se trabalhar com a bancada, apesar de não se ter entrado em detalhes de como funciona o servomotor, *drive* e outros componentes.

O foco principal era fechar a malha, o que foi possível, conforme se observa na seção 3.4. Agora, pode-se testar mais técnicas de controle em malha fechada com a câmera, o que não foi feito antes nesta bancada. Apesar do resultado de um controlador P não ser ideal, é o verdadeiro início da validação de controladores para *risers* com esta bancada, pois na realidade *ROVs* com câmeras são utilizados, o que justifica também ser utilizada uma câmera aqui.

4.1 Perspectivas Futuras

Pra o Trabalho de Graduação 2, objetiva-se utilizar uma técnica de controle mais rebuscada, utilizando-se o projeto de Fabrício et al [4]. Nele, o modelo é levado em consideração através de redução modal, que o deixa mais simples em questão de custo computacional. Após verificar-se se a mencionada técnica de controle funciona bem mesmo com perturbações, pretende-se analisar outras literaturas na área de controle preditivo e tentar projetar um controlador novo que funcione melhor que o anterior, em algum aspecto.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] FORTALEZA, E.; ALBUQUERQUE, D.; YAMAMOTO, M. An investigation about the trajectory control during the subsea equipment installation using cable. In: *Volume 1: Offshore Technology*. ASME International, 2012. Disponível em: <<http://dx.doi.org/10.1115/OMAE2012-83798>>.
- [2] SOUSA, R. B. *Implementação de Controle de Riser, Validação Experimental e Análise Através de Processamento de Imagens*. Brasília, DF: Faculdade de Tecnologia, Universidade de Brasília, 2015. Trabalho de Graduação em Engenharia de Controle e Automação.
- [3] PETROBRAS. *Pré-Sal: Exploração e Produção de Petróleo*. Disponível em: <http://www.petrobras.com.br/pt/nossas-atividades/areas-de-atuacao/exploracao-e-producao-de-petroleo-e-gas/pre-sal/>. Acesso em: 27/11/2015.
- [4] MONTEIROS, F. R.; FILHO, J. O. de A. L.; FORTALEZA, E. Modal reduction based tracking control for installation of subsea equipments*. *IFAC-PapersOnLine*, v. 48, n. 6, p. 15 – 20, 2015. ISSN 2405-8963. 2nd {IFAC} Workshop on Automatic Control in Offshore Oil and Gas Production {OOGP} 2015 Florianópolis, Brazil, 27–29 May 2015. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2405896315008708>>.
- [5] SERCOS INTERNATIONAL. *Introduction to Sercos*. Disponível em: <http://www.sercos.com/technology/index.htm>. Acesso em: 28/11/2015.
- [6] ALLEN-BRADLEY. *User Manual Kinetix 6000 Multi-axis Servo Drives*. July 2015. Disponível em: http://literature.rockwellautomation.com/idc/groups/literature/documents/um/2094-um001_en-p.pdf. Rockwell Automation Publication 2094-UM001I-EN-P. Acesso em: 28/11/2015.
- [7] KOLLMORGEN. *Servomotors*. Disponível em: <http://www.kollmorgen.com/en-us/products/motors/servo/servomotors/>. Acesso em: 28/11/2015.
- [8] RTA AUTOMATION. *DeviceNet™Unplugged – A View “Under the Hood” for End Users*. Disponível em: <http://www.rtaautomation.com/technologies/devicenet/>. Acesso em: 30/11/2015.
- [9] ROCKWELL AUTOMATION. *DeviceNet Network*. Disponível em: <http://ab.rockwellautomation.com/Networks-and-Communications/DeviceNet-Network>. Acesso em: 30/11/2015.

- [10] ODVA. *Ethernet/IP - Quick Start for Vendors Handbook*. Disponível em: https://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00213R0_EtherNetIP_Developers_Guide.pdf. Acesso em: 30/11/2015.
- [11] RINALDI, J. *The 6.5 Things You Must Know about EtherNet/IP*. Disponível em: <http://www.rtaautomation.com/technologies/ethernetip/>. Acesso em: 30/11/2015.
- [12] ROCKWELL AUTOMATION. *A Single IT-Friendly Network for Enterprise and Industrial Applications*. Disponível em: <http://www.rockwellautomation.com/global/products-technologies/integrated-architecture/ethernet-ip.page>. Acesso em: 30/11/2015.
- [13] EPUSP. *Automação elétrica de processos industriais*. Disponível em: http://lara.unb.br/~gaborges/disciplinas/ca/teoria_final_1_prova.pdf. Acesso em: 29/11/2015.
- [14] BANNER ENGINEERING. *PresencePLUS Industrial Ethernet User's Guide Vol 2*. September 2015. Disponível em: <http://www.bannerengineering.com/en-US/products/sub/78#ui-tabs-37>. Acesso em: 29/11/2015.

ANEXOS

I. PROGRAMAS UTILIZADOS

I.1 Texto Estruturado

I.1.1 Inicialização do primeiro teste de malha aberta

```
/* Universidade de Brasilia  
Trabalho de Graduacao em Engenharia Mecatronica  
Alunos:  
Ataias Pereira Reis 10/0093817  
Emanuel Pereira Barroso Neto 11/0115716  
This code must be executed only once. */  
  
MSO(drive_axis ,MSO_1);  
  
speed [0] := 0.0;  
speed [1] := 0.0;  
speed [2] := 0.0;  
speed [3] := 0.0;  
speed [4] := 0.0;  
speed [5] := 0.0;  
speed [6] := 0.0;  
speed [7] := 0.0;  
speed [8] := 0.0;  
speed [9] := 0.0;  
speed [10] := 0.0;  
speed [11] := 0.0;  
speed [12] := 0.0;  
speed [13] := 0.0;  
speed [14] := 0.0;  
speed [15] := 0.0;  
speed [16] := 0.0;  
speed [17] := 0.0;  
speed [18] := 0.0;  
speed [19] := 0.0;  
speed [20] := 0.0;  
speed [21] := 0.0147026554205;  
speed [22] := 0.0547055820855;  
speed [23] := 0.114361720023;  
speed [24] := 0.18111091036;
```

```
speed [25] := 0.251809893147;
speed [26] := 0.32607467121;
speed [27] := 0.403453064793;
speed [28] := 0.483471620607;
speed [29] := 0.565658044965;
speed [30] := 0.649536805548;
speed [31] := 0.734630464537;
speed [32] := 0.820458741642;
speed [33] := 0.906543473223;
speed [34] := 0.992424061587;
speed [35] := 1.07765619092;
speed [36] := 1.16189824235;
speed [37] := 1.24456943015;
speed [38] := 1.32036725857;
speed [39] := 1.3812552175;
speed [40] := 1.43231065067;
speed [41] := 1.48663214948;
speed [42] := 1.54839415747;
speed [43] := 1.61599008767;
speed [44] := 1.68757723373;
speed [45] := 1.7613579514;
speed [46] := 1.83569169336;
speed [47] := 1.90907847349;
speed [48] := 1.9801747816;
speed [49] := 2.04784594244;
speed [50] := 2.11113619377;
speed [51] := 2.16944638193;
speed [52] := 2.22182598313;
speed [53] := 2.26825099412;
speed [54] := 2.30985215012;
speed [55] := 2.34320587986;
speed [56] := 2.36620871737;
speed [57] := 2.38071882377;
speed [58] := 2.38672834648;
speed [59] := 2.38430263817;
speed [60] := 2.37407699776;
speed [61] := 2.35619724752;
speed [62] := 2.33055863496;
speed [63] := 2.29754175796;
speed [64] := 2.25797832185;
speed [65] := 2.21262636405;
speed [66] := 2.16123527696;
```

```

speed [67] := 2.10467481035;
speed [68] := 2.04388225087;
speed [69] := 1.97863997419;
speed [70] := 1.90950133568;
speed [71] := 1.83693203375;
speed [72] := 1.7619821587;
speed [73] := 1.686412295;
speed [74] := 1.60932227957;
speed [75] := 1.53031257908;
speed [76] := 1.45140199977;
speed [77] := 1.37297157979;
speed [78] := 1.29394180319;
speed [79] := 1.2157528212;
speed [80] := 1.1408355106;
speed [81] := 1.06779007087;
speed [82] := 0.995420551475;
speed [83] := 0.924581999131;
speed [84] := 0.855323010933;
speed [85] := 0.788477897412;
speed [86] := 0.726538190521;
speed [87] := 0.66801109369;
speed [88] := 0.607741032753;
speed [89] := 0.547567758592;
speed [90] := 0.493301637758;
speed [91] := 0.440419162243;
speed [92] := 0.387507265216;
speed [93] := 0.338400915621;
speed [94] := 0.291658541896;
speed [95] := 0.248512542153;
speed [96] := 0.209796973955;
speed [97] := 0.172530018125;
speed [98] := 0.141664432169;
speed [99] := 0.11599366767;
speed [100] := 0.087107967923;
speed [101] := 0.0627295944751;
speed [102] := 0.0490272896769;
speed [103] := 0.0374349286012;
speed [104] := 0.0234408281723;
speed [105] := 0.0103613796074;
speed [106] := 0.0;
speed [107] := 0.0;
speed [108] := 0.0;

```

```
speed[109] := 0.0;
speed[110] := 0.0;
speed[111] := 0.0;
speed[112] := 0.0;
speed[113] := 0.0;
speed[114] := 0.0;
speed[115] := 0.0;
speed[116] := 0.0;
speed[117] := 0.0;
speed[118] := 0.0;
speed[119] := 0.0;
speed[120] := 0.0;
speed[121] := 0.0;
speed[122] := 0.0;
speed[123] := 0.0;
speed[124] := 0.0;
speed[125] := 0.0;
speed[126] := 0.0;
speed[127] := 0.0;
speed[128] := 0.0;
speed[129] := 0.0;
speed[130] := 0.0;
speed[131] := 0.0;
speed[132] := 0.0;
speed[133] := 0.0;
speed[134] := 0.0;
speed[135] := 0.0;
speed[136] := 0.0;
speed[137] := 0.0;
speed[138] := 0.0;
speed[139] := 0.0;
speed[140] := 0.0;
speed[141] := 0.0;
speed[142] := 0.0;
speed[143] := 0.0;
speed[144] := 0.0;
speed[145] := 0.0;
speed[146] := 0.0;
speed[147] := 0.0;
speed[148] := 0.0;
speed[149] := 0.0;
speed[150] := 0.0;
```

```

speed[151] := 0.0;
speed[152] := 0.0;
speed[153] := 0.0;
speed[154] := 0.0;
speed[155] := 0.0;
speed[156] := 0.0;
speed[157] := 0.0;
speed[158] := 0.0;
speed[159] := 0.0;
speed[160] := 0.0;
speed[161] := 0.0;
speed[162] := 0.0;
speed[163] := 0.0;
speed[164] := 0.0;
speed[165] := 0.0;
speed[166] := 0.0;
speed[167] := 0.0;
speed[168] := 0.0;
speed[169] := 0.0;
speed[170] := 0.0;

dataInitialized := 1;

```

I.1.2 Inicialização do segundo teste de malha aberta

```

/* Universidade de Brasilia
Trabalho de Graduacao em Engenharia Mecatronica
Alunos:
Ataiaas Pereira Reis 10/0093817
Emanuel Pereira Barroso Neto 11/0115716
This code must be executed only once. */

```

```
MSO(drive_axis,MSO_1);
```

```

speed[0] := 0.0;
speed[1] := 0.0;
speed[2] := 0.0;
speed[3] := 0.0;
speed[4] := 0.0;
speed[5] := 0.0;
speed[6] := 0.0;
speed[7] := 0.0;
speed[8] := 0.0;

```

```

speed [9] := 0.0;
speed [10] := 0.0;
speed [11] := 0.0;
speed [12] := 0.0;
speed [13] := 0.0;
speed [14] := 0.0;
speed [15] := 0.0;
speed [16] := 0.0;
speed [17] := 0.0;
speed [18] := -0.0692113524459;
speed [19] := 0.314942000684;
speed [20] := 1.64847179452;
speed [21] := 3.06203932423;
speed [22] := 3.88556305076;
speed [23] := 4.25984881416;
speed [24] := 4.24772217569;
speed [25] := 3.9166637819;
speed [26] := 3.35019221407;
speed [27] := 2.64853848716;
speed [28] := 1.92086693496;
speed [29] := 1.27214274157;
speed [30] := 0.788466188462;
speed [31] := 0.524290908189;
speed [32] := 0.493490156489;
speed [33] := 0.694837969193;
speed [34] := 0.822419246039;
speed [35] := 0.418206857379;
speed [36] := 0.0262247526643;
speed [37] := 0.230482402967;
speed [38] := 0.789536874806;
speed [39] := 1.48057490162;
speed [40] := 2.15097967763;
speed [41] := 2.69969451667;
speed [42] := 3.0612889055;
speed [43] := 3.19935866684;
speed [44] := 3.10564873155;
speed [45] := 2.7984308071;
speed [46] := 2.32194132411;
speed [47] := 1.78925311378;
speed [48] := 1.23475193349;
speed [49] := 0.680977178876;
speed [50] := 0.269581981875;

```

```

speed[51] := 0.0735006589406;
speed[52] := 0.0;
speed[53] := -0.0411084579085;
speed[54] := -0.0752386045019;
speed[55] := -0.0939490510793;
speed[56] := -0.0958419020537;
speed[57] := -0.0879379646741;
speed[58] := -0.0665110466718;
speed[59] := -0.0340396907113;
speed[60] := 0.0;
speed[61] := 0.0116207365686;
speed[62] := 0.0239437089848;
speed[63] := 0.0232352765649;
speed[64] := 0.0;
speed[65] := 0.0;
speed[66] := 0.0;
speed[67] := 0.0;
speed[68] := 0.0;
speed[69] := 0.0;
speed[70] := 0.0;
speed[71] := 0.0;
speed[72] := 0.0;
speed[73] := 0.0;
speed[74] := 0.0;
speed[75] := 0.0;
speed[76] := 0.0;
speed[77] := 0.0;
speed[78] := 0.0;
speed[79] := 0.0;
speed[80] := 0.0;
speed[81] := 0.0;
speed[82] := 0.0;
speed[83] := 0.0;
speed[84] := 0.0;
speed[85] := 0.0;

dataInitialized := 1;

```

I.1.3 Inicialização dos testes de malha fechada

```

/* Universidade de Brasilia
Trabalho de Graduacao em Engenharia Mecatronica
Alunos :

```

Ataias Pereira Reis 10/0093817
 Emanuel Pereira Barroso Neto 11/0115716
 This code must be executed only once. */

```
MSO(drive_axis,MSO_1);
Kp := 0.0075;
```

```
dataInitialized := 1;
```

I.1.4 Inicialização da rede DeviceNET

```
Local:2:O.CommandRegister.Run [:=] 1;
if NOT dataInitialized then
    EVENT(Initialize_speed);
end_if;
```

I.1.5 Programa de controle em malha aberta

```
// k is the index of the speed vector
if dataInitialized AND k < 85 then
    MAJ(drive_axis,MAJ_1,0,speed[k],0,50.0,1,50.0,1,0,50.0,50.0,1,0,0);
    k:=k + 1;
end_if;

if k >= 85 then
    MAS(drive_axis,MAS_1,0,0,100,1,0,100,1);
    if MAS_1.DN then
        MSF(drive_axis,MSF_0);
    end_if;
end_if;
```

I.1.6 Programa de controle proporcional - malha fechada

```
// k is the index of the speed vector

if dataInitialized AND k < 300 then
    error := 300.0 - position;
    speed := error*Kp;
    MAJ(drive_axis,MAJ_1,0,speed,0,50.0,1,50.0,1,0,50.0,50.0,1,0,0);
    k:=k + 1;
end_if;

if k >= 300 then
    MAS(drive_axis,MAS_1,0,0,100,1,0,100,1);
```

```

if MAS_1.DN then
    MSF( drive_axis ,MSF_0 );
end_if;
end_if;

```

I.2 Linguagem *ladder*

I.2.1 Execução de *trigger* da câmera

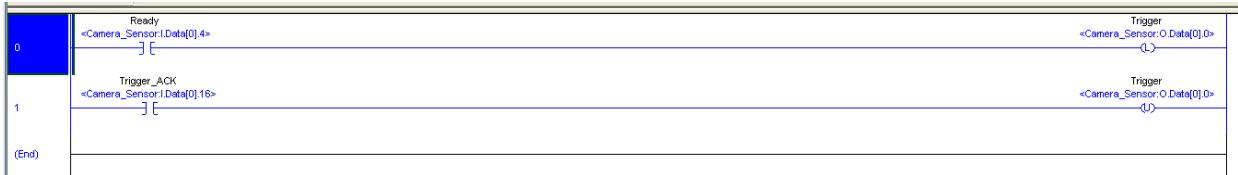


Figura I.1: *Trigger* da câmera

I.2.2 Rotina de parada de emergência

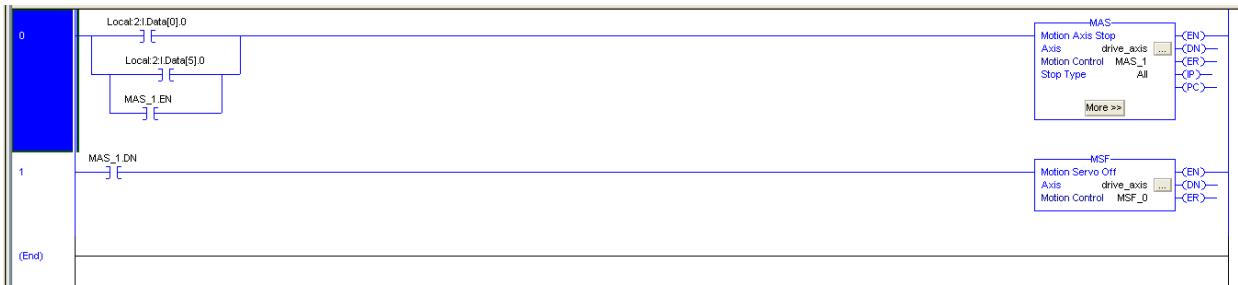


Figura I.2: Parada de emergência

I.3 Programas da câmera

I.3.1 Detecção da posição horizontal da bolinha

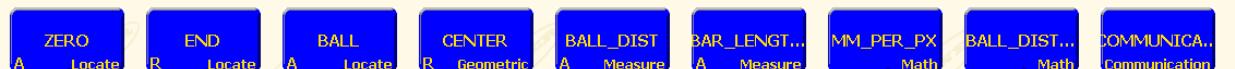
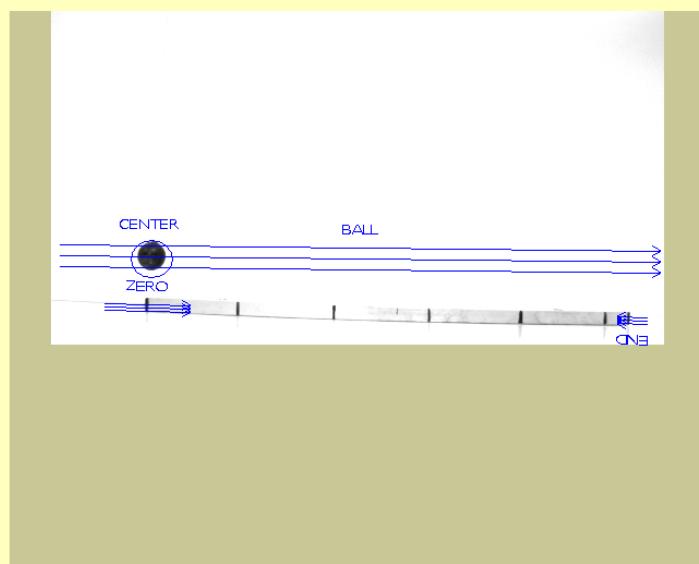


Figura I.3: Detecção da posição horizontal da bolinha