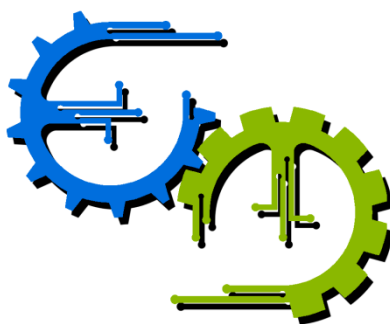


# **TRABALHO DE GRADUAÇÃO**

## **IMPLEMENTAÇÃO DE CONTROLE DE RISER, VALIDAÇÃO EXPERIMENTAL E ANÁLISE ATRAVÉS DE PROCESSAMENTO DE IMAGENS**

**Rédytton Brenner Sousa**

**Brasília, 04 de Fevereiro de 2015**



**ENGENHARIA  
MECATRÔNICA**  
UNIVERSIDADE DE BRASÍLIA

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

## TRABALHO DE GRADUAÇÃO

# IMPLEMENTAÇÃO DE CONTROLE DE RISER, VALIDAÇÃO EXPERIMENTAL E ANÁLISE ATRAVÉS DE PROCESSAMENTO DE IMAGENS

**Rédytton Brenner Sousa**

Relatório submetido como requisito parcial para obtenção  
do grau de Engenheiro de Controle e Automação.

### **Banca Examinadora**

Prof. Eugênio Liborio Feitosa Fortaleza, UnB/ ENM (Orientador)	_____
Prof. Eduardo Stockler Tognetti, UnB/ CIC (Co-orientador)	_____
Prof. José Alfredo Ruiz Vargas, UnB/ENE (Examinador interno)	_____
Eng. Léo Vitor Learth, Radix – Engenharia e Software (Examinador externo)	_____

Brasília, 04 de Fevereiro de 2015

## FICHA CATALOGRÁFICA

RÉDYTTON, BRENNER SOUSA

Implementação de controle de riser, validação experimental e análise através de processamento de imagens

[Distrito Federal] 2014.

xi, 49p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2014).

Trabalho de Graduação – Universidade de Brasília - Faculdade de Tecnologia.

- |                                   |   |
|-----------------------------------|---|
| 1. Controle de risers             | 2. Planejamento de trajetória - <i>flatness</i> |
| 3. Controlador lógico programável | 4. Processamento de imagens                     |

I. Mecatrônica/FT/UnB

II. Implementação de controle de riser, validação experimental e análise através de processamento de imagens

## REFERÊNCIA BIBLIOGRÁFICA

SOUSA, R. B., (2014). IMPLEMENTAÇÃO DE CONTROLE DE RISER, VALIDAÇÃO EXPERIMENTAL E ANÁLISE ATRAVÉS DE PROCESSAMENTO DE IMAGENS. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 21, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 49p.

## CESSÃO DE DIREITOS

AUTOR: Rédytton Brenner Sousa

TÍTULO DO TRABALHO DE GRADUAÇÃO: Implementação de controle de riser, validação experimental e análise através de processamento de imagens.

GRAU: Engenheiro

ANO: 2014

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 do autor.

---

Rédytton Brenner Sousa  
SHCN EQ 406/407 BLOCO A - ASA NORTE  
BRASÍLIA – DF

## AGRADECIMENTOS

Primeiramente, agradeço a Deus pelas oportunidades e por ser o alicerce da minha família.

Agradeço aos meus pais, Sirlene e Rafael, que com muito carinho e dedicação me criaram. Ensinaram-me sobre justiça, lealdade, determinação, paciência, humildade e sobre o amor, me mostrando isso a cada dia. Contaram-me sobre a vida, sobre as verdades e as mentiras, e me ensinaram a viver. Se cheguei onde estou e se tive sucesso em algo, foi por e para vocês.

Agradeço a minha irmã, Rayssa Suel, que é inspiração pros meus dias difíceis. Amiga fiel, que sei que poderei contar sempre que eu precisar. Agradeço aos meus avós que, com muita doçura, cuidaram de mim quando eu era novo e, até hoje, me ensinam coisas novas, inclusive sobre engenharia.

Agradeço a Radix e ao INPE pelas grandes oportunidades oferecidas a mim durante os estágios, mesmo sem que eu tenha muito a oferecer. Ao Léo Vitor, por me dar direção conhecimento e motivação durante minha estadia no Rio.

Agradeço aos professores Edgar Nobuo Mamiya, Eugênio Libório Feitosa Fortaleza e Eduardo Stockler Tognetti, meus mestres, que me orientaram durante a graduação com muita paciência.

Agradeço a Mecajun – Mecatrônica Junior de Brasília, que se tornou minha segunda casa durante dois anos e os colegas, hoje chamados de amigos, que sempre pude contar.

Aos amigos da república, Eduardo, Marlon e Leonardo, que toparam o desafio de sair de casa para estudar em outro estado. Pelas noites que passamos assistindo filmes e séries, fazendo churrasco ou estudando para as provas. Vocês fazem, sem dúvida, parte da minha vida e espero preservar essa amizade pra sempre.

Aos amigos de Goiânia que, mesmo com todo o stress que eu tinha, me recebiam com um abraço confortante.

E ao Luan Haickel Araújo, que além de ser um ótimo amigo me ajudou na conclusão deste trabalho.

*Rédytton Brenner Sousa.*

## RESUMO

As operações de petróleo em alto mar tem se tornado mais comuns a cada dia. Elas possuem grandes desafios tecnológicos e altos custos. Durante a operação, os *risers* são responsáveis por possibilitar um caminho para a subida dos hidrocarbonetos do fundo mar até a plataforma. Além disso, os risers são utilizados durante a perfuração, manutenção e instalação de equipamentos nos poços de petróleo. Durante a manutenção, o riser é desconectado do poço, para a instalação de equipamentos, por exemplo, e em seguida se inicia a operação de reentrada no poço. O deslocamento do *riser* ocorre a partir do posicionamento da plataforma, de modo é demandado tempo e alto custo operacional para ser concluída. Muitos dos equipamentos utilizados são alugados, o que torna a velocidade das operações um fator crucial para a indústria *offshore*. Neste contexto, o trabalho apresenta um controle de trajetória para a plataforma, de modo a posicionar a ponta do *riser* no local desejado. Além disso, mostra um experimento em escala laboratorial para validar as trajetórias propostas utilizando sensores e CLPs conhecidos na indústria.

**Palavras-chave:** Controle de Riser, Planejamento de Trajetória, Identificação por Processamento de Imagens, Controlador Lógico Programável (CLP).

## ABSTRACT

Petroleum offshore operations have become more common every day. They have technical challenges and high costs. During operation, the risers are responsible for enabling a path to rising oil from the sea bottom to the platform. Furthermore, the risers are used during drilling and installation of equipment in oil wells. During maintenance, the riser is disconnected from the pit for the installation of equipment, for example, and then starts re-entry operation in the well. The displacement of the riser occurs from the platform position, so is demanded time and high operating cost to complete. Many of the equipment used are rented, which makes the speed of operations a crucial factor for the offshore industry. In this context, the paper presents a path control to the platform, so as to position the tip of the riser at the desired location. It also shows an experiment in laboratory scale to validate the paths proposed using sensors and PLCs known in the industry.

**Keywords:** Riser Control, Trajectory Planning, Identification by Image Processing, Programmable Logic Controller (PLC).

# SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>1</b>
1.1. CONTEXTUALIZAÇÃO E DEFINIÇÃO DO PROBLEMA	1
1.2. OBJETIVOS	3
1.3. APRESENTAÇÃO DO MANUSCRITO	4
<b>2. FUNDAMENTOS TEÓRICOS</b>	<b>5</b>
2.1. MODELAGEM	5
2.2. FORÇAS HIDRODINÂMICAS	5
2.3. CÁLCULO DE $T(Z)$	7
2.4. PLANEJAMENTO DE TRAJETÓRIA	7
2.5. DIFERENÇAS FINITAS, DECOMPOSIÇÃO E REDUÇÃO MODAL	8
2.6. CÁLCULO DA SAÍDA PLANA	9
2.7. GERANDO AS TRAJETÓRIAS	11
<b>3. FUNDAMENTOS TÉCNICOS E APLICAÇÃO</b>	<b>13</b>
3.1. EXPERIMENTO	13
3.2. REDES INDUSTRIAIS	13
3.3. SERCOS	17
3.4. DEVICENET	18
3.5. CONTROLADOR LÓGICO PROGRAMÁVEL	19
3.6. LINE INTERFACE MODULE	23
3.7. SERVO DRIVE	24
3.8. SERVO MOTOR	28
3.9. SENSORES INDUTIVOS	28
3.10. ENCODERS	29
3.11. CÂMERA PRESENCEPLUS	30
3.12. PROGRAMA LADDER	31
3.13. CRIANDO O PERFIL DE VELOCIDADE	35
<b>4. AQUISIÇÃO E PROCESSAMENTO DE IMAGENS, ANÁLISE E CONCLUSÃO</b>	<b>39</b>
4.1. INTRODUÇÃO	39
4.2. CALIBRAÇÃO DA CÂMERA	40
4.3. AQUISIÇÃO E PROCESSAMENTO DAS IMAGENS	41
4.4. ANÁLISE DOS RESULTADOS	44
4.5. CONCLUSÃO	47
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>48</b>

# LISTA DE FIGURAS

1.1	Operação de reentrada do riser.....	1
1.2	Método atual para reconexão ao poço .....	2
1.3	Proposta do trabalho para reentrada automatizada.....	2
2.1	Trajetória do carro e da ponta do barbante .....	12
3.1	Bancada utilizada para o experimento.....	13
3.2	Camadas do modelo OSI e suas funções .....	14
3.3	Topologia ponto a ponto.....	14
3.4	Topologia de barramento .....	15
3.5	Topologia em anel.....	15
3.6	Topologia em estrela.....	15
3.7	Topologia em árvore .....	16
3.8	Pares de fios trançados.....	16
3.9	Cabo coaxial .....	17
3.10	Fibra óptica multimodo .....	17
3.11	Cartão DeviceNet modelo 1756-DNM .....	19
3.12	Ciclo de scan do CLP .....	20
3.13	Controlador modelo 1756-L60M03SE .....	21
3.14	Interface do software RSLinx .....	22
3.15	Interface do software RSLogix500.....	22
3.16	Ferramenta Trend do RSLogix5000 .....	23
3.17	Line interface module modelo 2094-AL09 da Allen Bradley.....	24
3.18	Driver Kinetix 6000 da Allen Bradley .....	24
3.19	Circuito para gerar o sinal PW .....	25
3.20	Sinal PWM (em azul) e valor médio do sinal (em vermelho).....	25
3.21	Circuito interno do servo drive .....	27
3.22	Servo motor modelo MPL-A310F-SJ22A.....	28
3.23	Sensor indutivo 871T-R8B18 .....	29
3.24	Exemplo de encoder incremental e absoluto .....	29
3.25	Câmera utilizada no projeto, da Banner Engineering.....	30
3.26	Placa de processamento da câmera .....	30
3.27	Exemplo de programa em Ladder .....	31
3.28	Diagrama elétrico de uma partida direta.....	31
3.29	Ladder para partida direta .....	32
3.30	Moxion Servo On – ligar o servo .....	32
3.31	Motion Axis Home – utilizado uma única vez, no início da movimentação, para selecionar o ponto inicial .....	32

3.32	Motion Change Dynamics – atua diretamente sobre o servo, realizando a sua movimentação de acordo com os parâmetros escolhidos .....	32
3.33	Configuração dos ganhos proporcional e integral no driver .....	33
3.34	Parâmetros limite para o servo.....	34
3.35	Gerando a derivada da trajetória de posição .....	35
3.36	Diminuindo a quantidade de pontos da trajetória.....	35
3.37	Perfil de velocidade .....	36
3.38	Leitura no Trend para o perfil de velocidade .....	37
3.39	Sobreposição da velocidade teórica e da medida com o encoder .....	38
3.40	Sobreposição com a curva da velocidade medida deslocada.....	38
4.1	Bancada em tons de cinza 8 bits.....	39
4.2	Imagem da bancada com o zoom utilizado .....	40
4.3	Imagem da bancada com o contraste utilizado.....	41
4.4	Exemplo da execução das ferramentas de localização .....	42
4.5	Ferramenta de localização geométrica.....	43
4.6	Teste das ferramentas de localização .....	43
4.7	Ponta do barbante durante a movimentação com velocidade constante .....	44
4.8	Comparação entre saída com o carro em velocidade constante e a simulada .....	45
4.9	Perfil de velocidade medido pela câmera .....	45
4.10	Comparação entre saída real e saída desejada .....	46



## **LISTA DE TABELAS**

1.1	Dados para a simulação em escala real.....	11
1.2	Dados pra simulação em escala laboratorial .....	12
1.3	Especificações técnicas do Kinetix 6000 .....	25

# LISTA DE SÍMBOLOS

## Símbolos Latinos

$A$	Área da seção transversal	$[m^2]$
$d_e$	Diâmetro externo	$[m]$
$d_i$	Diâmetro interno	$[m]$
$E$	Módulo de Young	$[Pa]$
$g$	Aceleração da gravidade	$[m/s^2]$
$g'$	Gravidade aparente	$[m/s^2]$
$J$	Segundo momento de área	$[m^4]$
$L$	Comprimento do riser/barbante	$[m]$
$m$	Soma da massa intrínseca e adicional	$[Kg]$
$m_s$	Densidade linear de massa	$[kg/m]$
$T$	Resultante de forças de tração	$[N]$
$v$	Velocidade relativa entre o corpo e o fluido	$[m/s]$
$V$	Volume do corpo submerso	$[m^3]$
$z$	Eixo vertical	$[m]$

## Símbolos Gregos

$\tau$	Coeficiente de amortecimento	$[s^{-1}]$
$\mu$	Viscosidade dinâmica	$[m^2/s^2]$
$\rho$	Densidade do fluido	$[m^3/Kg]$
$\gamma$	Deslocamento horizontal	$[m]$
$w_1$	Frequência natural associada ao primeiro harmônico	$[rad/s]$
$w_2$	Frequência natural associada ao segundo harmônico	$[rad/s]$

## Grupos Adimensionais

$C_d$	Coeficiente de arrasto
$C_m$	Coeficiente de massa adicional
$f^*$	Trajetoória imposta
$n$	Número genérico de elementos
$u^*$	Trajetoória de entrada de referência
$y^*$	Trajetoória desejada para a saída
$\sigma_1$	Coeficiente de amortecimento associado ao primeiro harmônico
$\sigma_2$	Coeficiente de amortecimento associado ao segundo harmônico

## **Sobrescritos**

- Primeira derivada temporal
- Segunda derivada temporal
- Terceira derivada temporal
- (n) N-ésima derivada temporal

## **Siglas**

CLP	Controlador Lógico Programável
CNC	Comando Numérico Computadorizado
DP	Dynamic Positioning – Posicionamento Dinâmico
EDP	Equação Diferencial Parcial
GPS	Global Positioning System – Sistema de Posicionamento Global
ISO	International Organization for Standardization – Organização Internacional de Normalização
MAH	Motion Axis Home
MCD	Motion Change Dynamics
MSO	Motion Servo On
OSI	Open Systems Interconnection – Interconexão de Sistemas Abertos
PID	Proporcional – Integral – Derivativo
PWM	Pulse Width Modulation – Modulação por Largura de Pulso
ROV	Remotely Operated Vehicle – Veículo Operado Remotamente
SFC	Sequential Flow Chart – Fluxograma Sequencial
UTP	Unshielded Twisted Pair – Par Trançado

# Capítulo 1

## INTRODUÇÃO

### 1.1 Contextualização e definição do problema

Com a descoberta no Pré-Sal, que promete produção diária superior a 1 milhão de barris de óleo em 2017 (Petrobrás [19]), a exploração de petróleo em alto mar tem se tornado cada dia mais comum no Brasil. Nesse tipo de exploração os desafios tecnológicos são grandes dada a complexidade das operações.

A lâmina d'água, camada de água entre a plataforma (geralmente flutuante) e o fundo do mar, chega a 2000 metros. Esta distância é ocupada pelos risers, tubos e cabos que conectam a plataforma ao poço, que são utilizados durante a perfuração, manutenção e instalação dos equipamentos, além de ser o meio que possibilita o transporte do petróleo até as plataformas.

Atualmente, a maioria das operações offshore requer embarcações equipadas com sistema de posicionamento dinâmico (DP – Dynamic Positioning). Este sistema é composto por propulsores e sensores de localização/orientação (GPS, anemômetros, giroscópios etc.) e é capaz de posicionar a embarcação com precisão.

A operação de reentrada do riser pode ser vista na Figura 1.1. Nesta operação o riser se aproxima do poço para se reconectar. A posição da ponta inferior do riser é controlada a partir do posicionamento do topo (plataforma) com o sistema de posicionamento dinâmico.

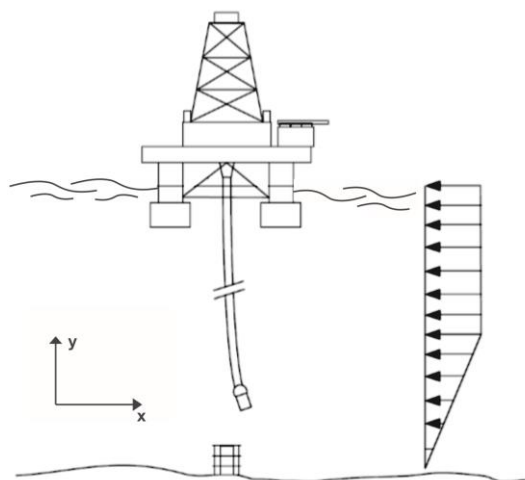


Figura 1.1 – Operação de reentrada do riser [3].

Atualmente, esta operação é feita através do operador da plataforma que, com o auxílio das imagens da extremidade inferior do riser capturadas por um ROV (Remotely Operated Vehicle), controla a plataforma através de um *joystick* utilizando o sistema de posicionamento dinâmico (Figura 1.2). Em águas profundas esta operação pode ser bastante complicada, pois no fundo do mar há dificuldades como ondas, correntezas, variações de salinidade e variações de temperatura e pressão. Além disso, o custo envolvido nas operações offshore é altíssimo e a segurança dos funcionários e dos equipamentos durante as operações é crucial.

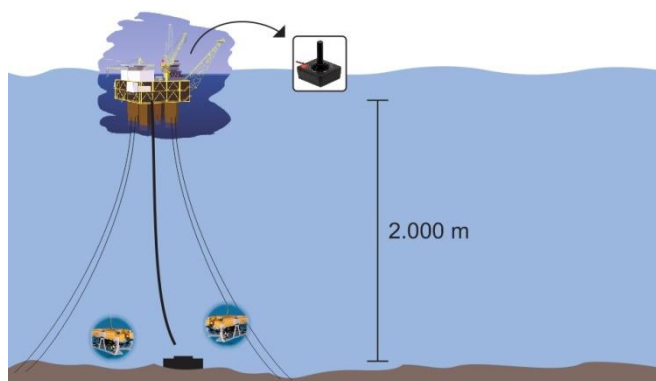


Figura 1.2 – Método atual para reconexão ao poço

É neste contexto que este trabalho se enquadra, propondo uma forma de realizar estas operações de forma mais rápida, eficiente, segura e automatizada (Figura 1.3), utilizando o planejamento de trajetória para a plataforma a partir da teoria de controle aplicada em um controlador e a validação experimental desses métodos.

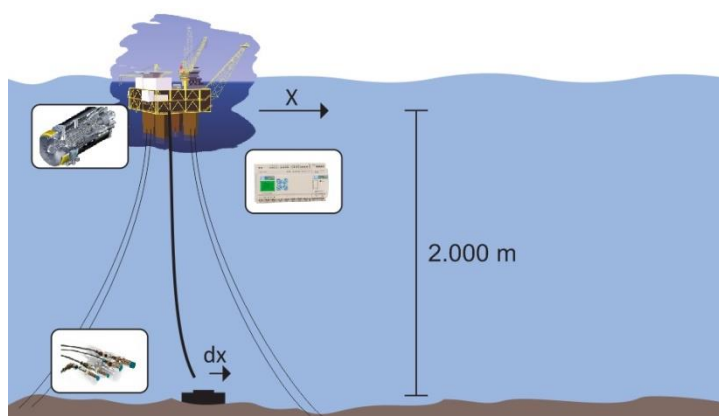


Figura 1.3 – Proposta do trabalho para reentrada automatizada

## 1.2 Objetivos

O trabalho visou, primeiramente, a adaptação do modelo dinâmico do comportamento do riser em ambiente marinho proposto por Fortaleza [3], para a escala laboratorial, utilizando um barbante de algodão no lugar do riser e ar no lugar da água, de modo a prever o comportamento da ponta do barbante a partir da atuação no topo. Posteriormente, adaptar uma bancada em escala laboratorial para validar experimentalmente o modelo adaptado.

Para medir o erro no experimento será utilizada uma câmera para gravar os deslocamentos e, posteriormente, será feito o processamento de imagens para quantizá-los. Desta forma, será possível comparar o trajeto esperado (teórico) e o realizado pelo experimento (prático).

### **1.3 Apresentação do manuscrito**

O capítulo 2 aborda a modelagem do riser e suas características específicas, além do conteúdo referente à técnica de controle para o planejamento de trajetória.

O capítulo 3 discorre sobre o experimento realizado, desde o estudo da planta e de seus componentes até a programação do CLP.

O capítulo 4 ilustra o método de aquisição e processamento das imagens. Também é feita a análise dos resultados obtidos, observações do autor e as propostas para trabalhos futuros.

# FUNDAMENTOS TEÓRICOS

### 2.1 Modelagem

Em geral, a estrutura responsável por mover os hidrocarbonetos do fundo do mar até as plataformas são os risers. Esses são geralmente cilíndricos e ocos, e podem atingir grandes comprimentos. São estruturas passivas e ficam acopladas à plataforma, que possui movimentação a partir do sistema de posicionamento dinâmico (DP).

O modelo governante do sistema é dado pela equação diferencial parcial (EDP) de quarta ordem a seguir:

$$m_s \frac{\partial^2 Y}{\partial t^2}(z, t) = -EJ \frac{\partial^4 Y}{\partial z^4}(z, t) + \frac{\partial}{\partial z} \left( T(z) \frac{\partial Y}{\partial z}(z, t) \right) + F_n(z, t) \quad (1)$$

Onde  $m_s$  é a densidade linear de massa do riser,  $E$  o seu módulo de Young,  $J$  o seu segundo momento de área,  $z$  é eixo vertical,  $t$  é o tempo e  $Y(z, t)$  é o deslocamento horizontal do riser.  $T(z)$  é uma função que descreve a resultante das forças trativas ao longo do comprimento do riser, que depende apenas de suas propriedades mecânicas.  $F_n(z, t)$  é a resultante das forças externas. Para uma coroa circular, como a secção transversal do riser,  $J$  é dado por:

$$J = \frac{\pi}{32} (d_e^4 - d_i^4) \quad (2)$$

Em que  $d_e$  e  $d_i$  são o diâmetro interno e o diâmetro externo do riser. A partir de agora, para simplificar a notação, omite-se a dependência de  $z$  e  $t$  em  $Y$  nas equações.

### 2.2 Forças hidrodinâmicas

Com exceção do topo e do fundo, as únicas forças externas atuantes sobre o riser são as forças hidrodinâmicas advindas do contato com o mar. Para representar essas forças é utilizada uma aproximação heurística dada por duas componentes, sendo elas o efeito do arrasto, que depende do quadrado da velocidade relativa entre o corpo e o fluido, e o efeito inercial, que depende do quadrado da aceleração [1]. A equação que relaciona essas componentes é chamada de Forças de Morison e pode ser visualizada a seguir:



$$F_n(z, t) = -c_m \rho V \frac{\partial^2 Y}{\partial t^2} - \frac{1}{2} c_d \rho A \left| \frac{\partial Y}{\partial t} \right| \frac{\partial Y}{\partial t} \quad (3)$$

Onde  $c_m$  e  $c_d$  são o coeficiente de massa adicional e o coeficiente de arrasto, respectivamente.  $V$  é o volume do corpo submerso,  $A$  a área da sessão de contato com o fluido e  $\rho$  a densidade do fluido. A dependência de interações transientes como variações locais na direção da velocidade do fluido fazem com que  $c_m$  e  $c_d$  variem com o tempo. Para o presente trabalho, dado que o formato do riser é cilíndrico, utiliza-se  $c_m = 1,2$  e  $c_d = 2$ , que são válidos para uma grande faixa de número de Reynolds [1].

Substituindo a equação (3) em (1) e fazendo  $m = m_s + c_m \rho V$  e  $v = \frac{c_d \rho A}{2m}$ , obtemos:

$$\frac{\partial^2 Y}{\partial t^2} = -\frac{EJ}{m} \frac{\partial^4 Y}{\partial z^4} + \frac{\partial}{\partial z} \left( \frac{T(z)}{m} \frac{\partial Y}{\partial z} \right) - v \left| \frac{\partial Y}{\partial t} \right| \frac{\partial Y}{\partial t} \quad (4)$$

Em estruturas com o comprimento muito maior que o diâmetro da secção transversal o segundo momento de inércia  $J$  tendem a ser muito pequenos. Logo, o termo  $\frac{EJ}{m} \frac{\partial^4 Y}{\partial z^4}$ , devido ao esforço interno por deflexão do riser, devido a sua rigidez, possui uma influência pequena em relação aos outros termos. A fim de simplificar o modelo temos a equação:

$$\frac{\partial^2 Y}{\partial t^2} = \frac{\partial}{\partial z} \left( \frac{T(z)}{m} \frac{\partial Y}{\partial z} \right) - v \left| \frac{\partial Y}{\partial t} \right| \frac{\partial Y}{\partial t} \quad (5)$$

Além disso, podemos linearizar o termo  $v \left| \frac{\partial Y}{\partial t} \right|$  referente ao atrito, substituindo-o por  $\tau$ , calculado com base em  $v$  e a velocidade média do riser. Desta forma, a linearização é precisa para a aplicação proposta, já que o deslocamento do riser é suave, com pequenas variações de velocidade.

O modelo passou a ser de segunda ordem, o que facilita a manipulação para aplicar as técnicas de controle e reduzir a quantidade de condições iniciais. Além disso, não possui mais o termo não linear do atrito, facilitando a manipulação algébrica. O novo modelo pode ser visto a seguir.

$$\frac{\partial^2 Y}{\partial t^2} = \frac{\partial}{\partial z} \left( \frac{T(z)}{m} \frac{\partial Y}{\partial z} \right) - \tau \frac{\partial Y}{\partial t} \quad (6)$$

### 2.3 Cálculo de $T(z)$

A operação de reentrada do riser é feita de duas formas, sendo elas com e sem carga na ponta. Quando a manobra é com carga, a tensão  $T(z)$  é, em geral, dependente apenas do peso do carregamento, visto que eles chegam a dezenas de toneladas, como a instalação do BOP (*Blowout Preventer*). Foi abordado apenas o caso sem massa na ponta, onde a tensão no riser depende de seu próprio peso e varia linearmente com  $z$ , como mostrado a seguir:

$$T(z) = (m_s - \rho A)gz \quad (7)$$

Onde  $g$  é a aceleração da gravidade. Substituindo a equação (7) em (6), temos:

$$\frac{\partial^2 Y}{\partial t^2} = \frac{\partial}{\partial z} \left( g' z \frac{\partial Y}{\partial z} \right) - \tau \frac{\partial Y}{\partial t} \quad (8)$$

Onde  $g' = \frac{(m_s - \rho A)g}{m}$ , chamada de gravidade aparente.

### 2.4 Planejamento de trajetória

Em geral, temos dois tipos de controle, seguir uma referência e eliminar perturbações. Muitas vezes, a referência não é um ponto de equilíbrio, mas sim uma trajetória arbitrária. Neste contexto podemos planejar uma trajetória para a entrada do sistema, de modo que a saída cumpra a trajetória desejada e, ainda, fazer o controle em malha fechada durante a execução da trajetória, para eliminar o efeito de possíveis perturbações e pequenos erros no planejamento da trajetória.

Assumindo um sistema de ordem  $n$ , representado por um conjunto de equações diferenciais:

$$\dot{x} = f(x, y), x \in \mathbb{R}^n, u \in \mathbb{R} \quad (9)$$

em que  $x$  é o vetor de estados  $n \times 1$ ,  $f$  é uma função vetorial  $n \times 1$  e  $u$  é a entrada do sistema. A saída deste sistema pode ser dada por:

$$y(t) = g(x, u) \quad (10)$$

em que  $\mathbf{g}$  é uma função vetorial arbitrária que depende dos estados do sistema e da entrada.

Deseja-se que a saída tenha um determinado comportamento, ou seja, que se comporte como uma trajetória desejada, chamada aqui de  $\mathbf{y}^*$  a partir de uma entrada planejada, chamada de  $\mathbf{u}^*$ . Para que isso seja possível o sistema deve ser diferencialmente plano. A planicidade (em inglês *flatness*) de um sistema é condicionada à possibilidade da parametrização de todas as variáveis do sistema em relação a um conjunto finito de variáveis “endógenas”, denominadas saídas planas, e um número finito de suas derivadas temporais. A dimensão das saídas planas é igual ao número de variáveis do sistema. Não há necessariamente somente um conjunto de saídas planas e, além disso, as saídas planas podem ser as próprias variáveis de estado do sistema ou combinações lineares delas.

Sendo satisfeita esta condição o planejamento e acompanhamento de trajetória se torna possível e, consideravelmente, simples de ser feito. Considerando um sistema genérico dado pela equação (9).

Caso esse sistema seja diferencialmente plano, podemos escrever uma função do vetor de estados  $\mathbf{x}$ , da entrada  $\mathbf{u}$  e suas derivadas, como se segue:

$$\Phi = \Phi(x, u, \dot{u}, \ddot{u}, \ddot{\ddot{u}}, \dots, u^n) \quad (11)$$

Tal que a saída plana  $\Phi$  parametrize completamente todas as variáveis do sistema. Desta forma, podemos representar os estados e a entrada em função da saída plana e suas derivadas temporais:

$$\begin{aligned} x &= A(\Phi, \dot{\Phi}, \ddot{\Phi}, \ddot{\ddot{\Phi}}, \dots, \Phi^n) \\ u &= B(\Phi, \dot{\Phi}, \ddot{\Phi}, \ddot{\ddot{\Phi}}, \dots, \Phi^{n+1}) \end{aligned} \quad (12)$$

## 2.5 Diferenças finitas, decomposição e redução modal

O riser é um sistema de parâmetros distribuídos. Uma implementação computacional exige um vetor de estados finito e para tal, é preciso discretizar espacialmente o sistema. Isto foi feito pela aproximação das derivadas parciais no espaço, por diferenças finitas, conforme em [6] e [7].

Modelos de sistemas de ordem muito alta ou, inclusive, de ordem infinita (como o riser) podem ser simplificados através de técnicas de redução de ordem. O objetivo é a redução dos graus de liberdade do sistema, de modo a não alterar as características dinâmicas do modelo, tornando possível a manipulação do sistema em baixa ordem. A decomposição modal é descrita em [4] e [5] e o método utilizado para a redução aplicada foi de acordo com [6]

## 2.6 Cálculo da saída plana

Considerando o modelo reduzido o sistema pode ser representado por:

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{pmatrix} = \begin{pmatrix} \sigma_1 & \omega_1 & 0 & 0 \\ -\omega_1 & \sigma_1 & 0 & 0 \\ 0 & 0 & \sigma_2 & \omega_2 \\ 0 & 0 & -\omega_2 & \sigma_2 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} u \quad (13)$$

Onde  $\sigma_1, \sigma_2, \omega_1$  e  $\omega_2$  são o coeficiente de amortecimento em relação ao primeiro e segundo harmônico e a frequência natural de oscilação em relação ao primeiro e segundo harmônico, respectivamente.

$$y = (c_1 \ c_2 \ c_3 \ c_4) \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} + (d)u \quad (14)$$

Verifica-se a controlabilidade do sistema pelo posto da matriz  $T$ , que deve ser igual a 4 e pode ser calculado por:

$$T = [B \ BA \ BA^2 \ BA^3] \quad (15)$$

A saída plana é uma combinação linear das variáveis de estado  $z_1, z_2, z_3$  e  $z_4$ , dada por:

$$f = [0 \ 0 \ 0 \ 1]T^{-1}z \quad (16)$$

A saída  $y$  do modelo reduzido é a posição do fundo do riser. Será feito o planejamento de trajetória para a saída  $y$  e, a partir dela, planejar uma trajetória para a saída plana. Se a saída  $y$  for substituída pela saída plana, o sistema deixa de ter ganho unitário. Pode-se calcular o novo ganho com a matriz de controlabilidade e planeja-se então, que a saída plana tenha valor final igual ao do seu ganho estático.

A partir da equação (17) podem-se calcular as derivadas temporais da saída plana. Para simplificar a notação serão atribuídos nomes aos coeficientes da saída plana e de suas derivadas, como se segue:

$$f = \alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 + \alpha_4 z_4 \quad (17)$$

Derivando  $f$  sucessivamente obtemos:

$$\begin{aligned}
 \dot{f} &= \alpha_5 z_1 + \alpha_6 z_2 + \alpha_7 z_3 + \alpha_8 z_4 \\
 \ddot{f} &= \alpha_9 z_1 + \alpha_{10} z_2 + \alpha_{11} z_3 + \alpha_{12} z_4 \\
 \ddot{f} &= \alpha_{13} z_1 + \alpha_{14} z_2 + \alpha_{15} z_3 + \alpha_{16} z_4 \\
 f^4 &= \alpha_{17} z_1 + \alpha_{18} z_2 + \alpha_{19} z_3 + \alpha_{20} z_4 + \gamma u
 \end{aligned} \tag{18}$$

Em que:

$$\begin{aligned}
 \alpha_5 &= \sigma_1 \alpha_1 - \omega_1 \alpha_2 \\
 \alpha_6 &= \sigma_1 \alpha_2 + \omega_1 \alpha_1 \\
 \alpha_7 &= \sigma_2 \alpha_3 - \omega_2 \alpha_4 \\
 \alpha_8 &= \sigma_2 \alpha_4 + \omega_2 \alpha_3 \\
 \alpha_9 &= \sigma_1 \alpha_5 - \omega_1 \alpha_6 \\
 \alpha_{10} &= \sigma_1 \alpha_6 + \omega_1 \alpha_5 \\
 \alpha_{11} &= \sigma_2 \alpha_7 - \omega_2 \alpha_8 \\
 \alpha_{12} &= \sigma_2 \alpha_8 + \omega_2 \alpha_7 \\
 \alpha_{13} &= \sigma_1 \alpha_9 - \omega_1 \alpha_{10} \\
 \alpha_{14} &= \sigma_1 \alpha_{10} + \omega_1 \alpha_9 \\
 \alpha_{15} &= \sigma_2 \alpha_{11} - \omega_2 \alpha_{12} \\
 \alpha_{16} &= \sigma_2 \alpha_{12} + \omega_2 \alpha_{11} \\
 \alpha_{17} &= \sigma_1 \alpha_{13} - \omega_1 \alpha_{14} \\
 \alpha_{18} &= \sigma_1 \alpha_{14} + \omega_1 \alpha_{13} \\
 \alpha_{19} &= \sigma_2 \alpha_{15} - \omega_2 \alpha_{16} \\
 \alpha_{20} &= \sigma_2 \alpha_{16} + \omega_2 \alpha_{15} \\
 \gamma &= b_1 \alpha_{13} + b_2 \alpha_{14} + b_3 \alpha_{15} + b_4 \alpha_{16}
 \end{aligned} \tag{19}$$

Com estas expressões é possível montar a seguinte equação matricial:

$$\begin{pmatrix} f \\ \dot{f} \\ \ddot{f} \\ \ddot{f} \\ f^4 \end{pmatrix} = M \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ u \end{pmatrix} \tag{20}$$

Onde

$$M = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & 0 \\ \alpha_5 & \alpha_6 & \alpha_7 & \alpha_8 & 0 \\ \alpha_9 & \alpha_{10} & \alpha_{11} & \alpha_{12} & 0 \\ \alpha_{13} & \alpha_{14} & \alpha_{15} & \alpha_{16} & 0 \\ \alpha_{17} & \alpha_{18} & \alpha_{19} & \alpha_{20} & \gamma \end{pmatrix} \tag{21}$$

A entrada  $u$  parametrizada pela saída flat e suas derivadas é:

$$u = [0 \ 0 \ 0 \ 0 \ 1]M^{-1} \begin{pmatrix} f \\ \dot{f} \\ \ddot{f} \\ \ddot{\ddot{f}} \\ f^4 \end{pmatrix} \quad (22)$$

$$u = \beta_0 f + \beta_1 \dot{f} + \beta_2 \ddot{f} + \beta_3 \ddot{\ddot{f}} + \beta_4 f^4$$

Com a equação (23) é possível calcular a trajetória de referência  $u^*(t)$  da entrada em função da trajetória imposta  $f^*(t)$  [2] para a saída plana.

$$f^*(t) = \begin{cases} 0, & t < 0 \\ \alpha_{11}t^{11} + \alpha_{10}t^{10} + \alpha_9t^9 + \alpha_8t^8 + \alpha_7t^7 + \alpha_6t^6 + \alpha_5t^5 + \alpha_4t^4 + \alpha_3t^3 + \alpha_2t^2 + \alpha_1t^1 + \alpha_0, & 0 \leq t < t_f \\ v_f, & t \geq t_f \end{cases} \quad (23)$$

Onde  $t_f$  é o tempo em que a trajetória se estabiliza no ganho estático da saída flat  $v_f$ . Os coeficientes  $\alpha_i$  são encontrados impondo valores iniciais e finais nulos até a quinta derivada.

## 2.7 Gerando as trajetórias

O algoritmo para criar as trajetórias nominais para a plataforma foi desenvolvido no software Matlab. A tabela 1.1 mostra os dados utilizados para as primeiras simulações [3]. Os dados são referentes à risers utilizados em outros trabalhos.

Tabela 1.1 – Dados para a simulação em escala real

Dados do riser	
Diâmetro externo - $d_e$	0.55 m
Diâmetro interno - $d_i$	0.5 m
Comprimento - L	2000 m
Módulo de Elasticidade - E	210 GPA
Densidade - $m_s$	7860 kg/m <sup>3</sup>
Dados do fluido	
Densidade do fluido - $\rho$	1000 kg/m <sup>3</sup>
Viscosidade dinâmica - $\mu$	10 <sup>3</sup> Pa.s

Neste caso os dados do fluido são referentes à água. No presente trabalho foram geradas trajetórias para o experimento montado em bancada. Os dados do riser foram substituídos pelos dados do barbante e os da água pelos do ar, conforme Tabela 1.2.

Tabela 1.2 – Dados para simulação em escala laboratorial

Dados do riser	
Diâmetro externo - $d_e$	0.161 m
Diâmetro interno - $d_i$	0 m
Comprimento - L	82 cm
Módulo de Elasticidade - E	2.1 MPA
Densidade - $m_s$	191 kg/m <sup>3</sup>
Dados do fluido	
Densidade do fluido - $\rho$	1.2754 kg/m <sup>3</sup>
Viscosidade dinâmica - $\mu$	17.2.10 <sup>-6</sup> Pa.s

Na Figura 2.1 é mostrada a saída do programa para as trajetórias geradas por *flatness*. Em verde é mostrada a posição da plataforma (referente ao servo motor) e em azul a ponta do riser (referente à ponta do barbante).

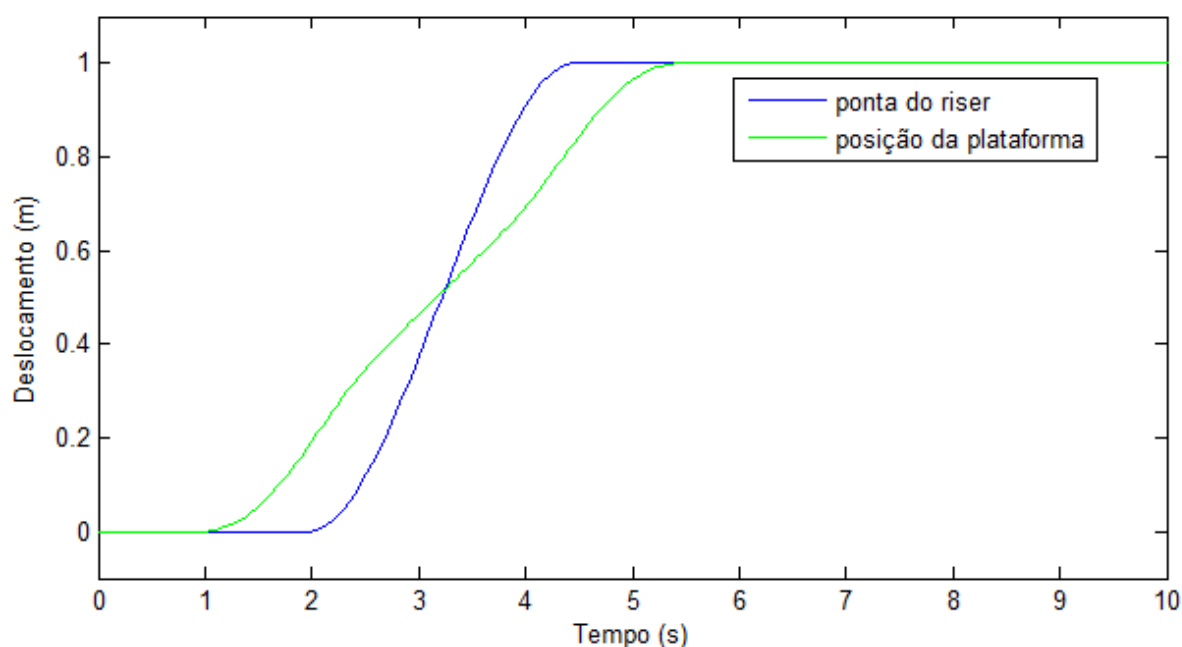


Figura 2.1 - Trajetória do carro e da ponta do barbante

O deslocamento escolhido foi de 1 metro, o mesmo utilizado nas simulações para as dimensões reais do conjunto plataforma/riser. O tempo para cumprir a trajetória é de cerca de 4 segundos. Vale ressaltar que o modelo perde precisão à medida que o tamanho do barbante é reduzido e para comprimentos inferiores a 70 cm a solução diverge.

# FUNDAMENTOS TÉCNICOS E APLICAÇÃO

### 3.1 Experimento

Um esboço da bancada utilizada pode ser vista na Figura 3.1. Esta bancada está localizada no laboratório de controle e automação da Universidade de Brasília e foi desenvolvida para a aplicação de leis de controle para pontes rolantes. A ponte rolante é um equipamento de elevação e transporte de carga que se movimenta apoiado sobre trilhos fixos. Um exemplo simples de controle que se assemelha a ponte rolante é o de um guindaste que carrega equipamentos pesados através de um cabo de aço.

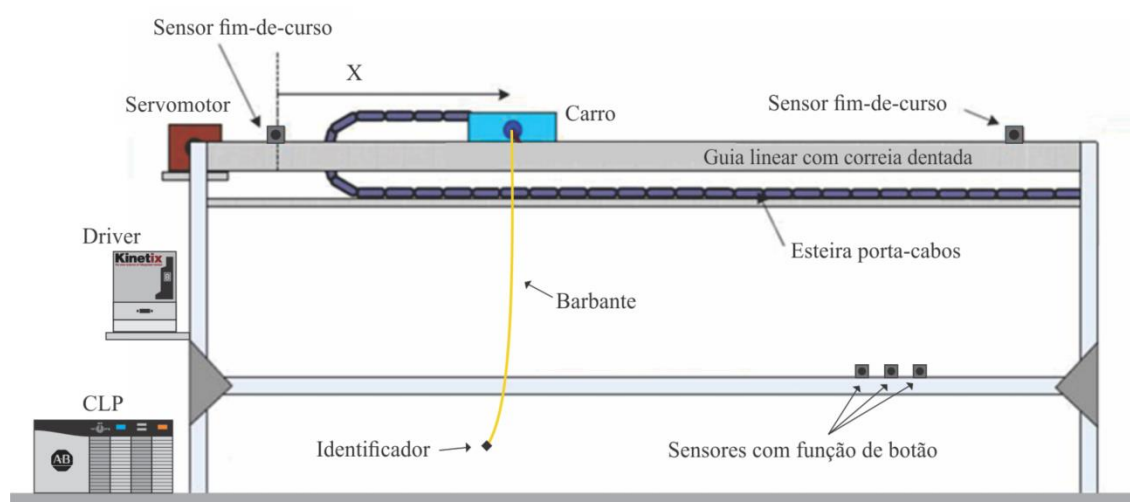


Figura 3.1 – Bancada utilizada para o experimento

Neste capítulo serão descritos os equipamentos e componentes utilizados bem como a rede para comunicação e a programação do CLP.

### 3.2 Redes industriais

Atualmente há necessidade de integrar máquinas e equipamentos têm crescido bastante na indústria. Além da praticidade de instalação e manutenção, os sistemas estão cada vez mais automatizados, o que exige um nível de controle mais elevado. O uso das redes industriais têm se tornado indispensável no ambiente industrial e corporativo.

Elas são utilizadas como estrutura de comunicação entre equipamentos e componentes com alta confiabilidade e eficiência na troca de dados. Com as redes é possível transmitir dados em altas taxas entre os diversos níveis de um sistema industrial, indo desde botões e sensores até controladores e computadores.



As redes são divididas por níveis e camadas. O nível mais baixo é responsável pela comunicação entre sensores e atuadores. O nível intermediário interliga CLPs e CNCs. O nível mais alto está relacionado a ferramentas de gestão, administração e gerenciamento.

Em relação às camadas, em geral, as redes são divididas em 7, definidas pela ISO (International Organization for Standardization - Organização Internacional para Padronizações) pelo modelo OSI (Open Systems Interconnection – Interconexão de Sistemas Abertos). As camadas foram definidas de forma que as camadas inferiores fornecem serviços para as camadas superiores, camadas superiores utilizam serviços das camadas inferiores.

As camadas padrões são: física, enlace (ou ligação), rede, transporte, sessão, apresentação e aplicação. A Figura 3.2 ilustra as 7 camadas e uma breve descrição de cada uma. Outros padrões utilizam um número inferior de camadas, como o TCP/IP que possui apenas 4 camadas, porém, com toda as funcionalidades das 7 camadas do padrão OSI.

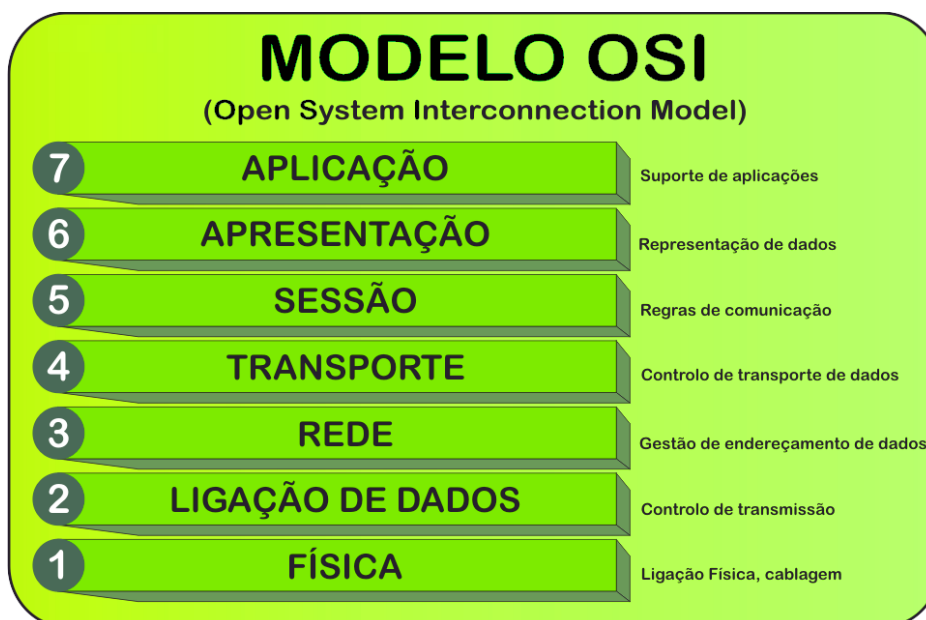


Figura 3.2 – Camadas do modelo OSI e suas funções [17].

Existem diversos tipos de redes e a escolha depende de diversos parâmetros tais como preço, facilidade de instalação, manutenção, topologia da rede, taxa de transmissão, algoritmo de acesso, meio físico de transmissão, barramento etc. Em relação à topologia física das redes temos os seguintes tipos:

Ponto a ponto: une dois dispositivos através de um meio de transmissão.

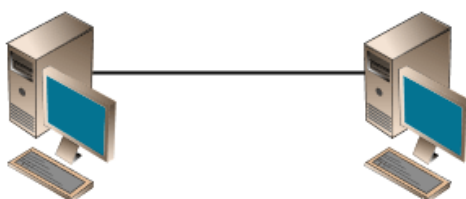


Figura 3.3 - Topologia ponto a ponto

Barramento: possui apenas um meio de transmissão onde são conectados todos os dispositivos que estarão em rede. Os dados trafegam para todos os dispositivos, mas somente o que possuir o endereço de destinatário irá ler a mensagem.

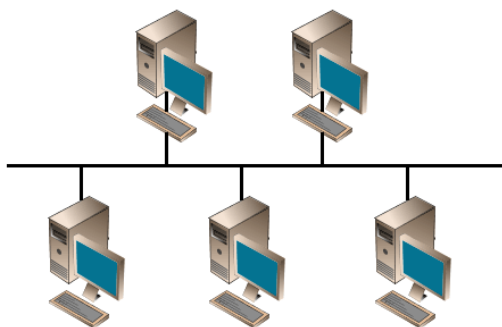


Figura 3.4 - Topologia de barramento

Anel: a conexão se dá de forma circular onde cada dispositivo atua como um repetidor do sinal. É simples de ser implementada, mas todo nó é um elo crítico.

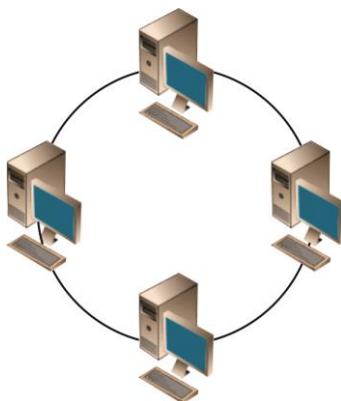


Figura 3.5 - Topologia em anel

Estrela: esta topologia possui um nó central que gerencia a transmissão de dados entre os nós. Este nós central geralmente é um switch, que fará o controle da taxa de transmissão e o endereçamento. Pode ser fisicamente igual à topologia de barramento, porém com lógica diferente.

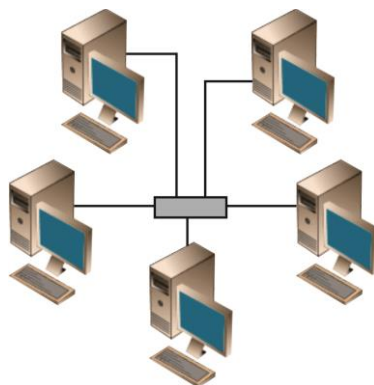


Figura 3.6 - Topologia em estrela

Árvore: é equivalente a várias redes em estrela interconectadas pelos nós centrais.

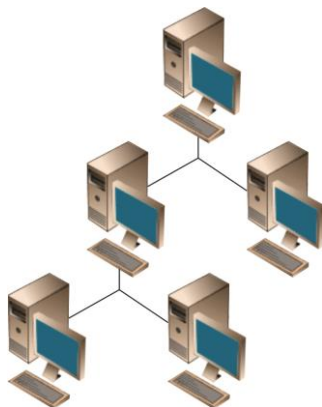


Figura 3.7 - Topologia em árvore

Híbrida e Gráfo: essas topologias são combinações das anteriores e, em geral, cada nó da rede contém uma rota alternativa para suprir eventuais falhas. Redes complexas e de grande porte costumam utilizar esses tipos de topologia.

O meio físico de transmissão pode ser através de cabos condutores ou de tecnologia sem fio. Apesar dos grandes avanços tecnológicos da transmissão sem fio (bluetooth, wireless, rádio, infravermelho etc) e dos protocolos de comunicação que utilizam este meio de transmissão (wireless hart, zigbee, wireless M-Bus, dentre outras), o meio físico mais utilizado na indústria é o meio cabeado. Dentre os cabos mais utilizados podemos destacar o par trançado, cabo coaxial e fibra óptica.

Par trançado: possui pares de fios entrelaçados um ao redor do outro de modo a cancelar as interferências eletromagnéticas. A qualidade da linha de transmissão depende, basicamente, da qualidade dos condutores, bitola dos fios, tipo da blindagem, distância a ser interligada e quantidade de ruído eletromagnético presente.

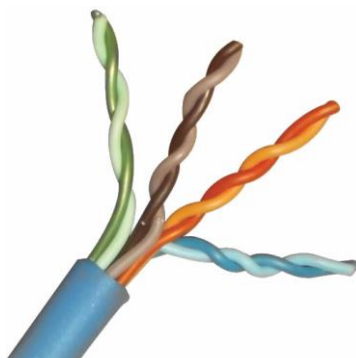


Figura 3.8 - Pares de fios trançados

Cabo coaxial: é utilizado em transmissões de alta velocidade e funciona através de impulsos elétricos. Possui um fio de cobre no centro, responsável por transmitir os dados, e um revestimento dielétrico juntamente com uma malha metálica, que eleva a tolerância a ruídos externos.

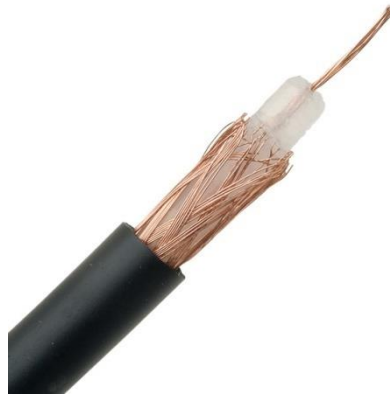


Figura 3.9 - Cabo coaxial

Fibra óptica: transmite dados através de impulsos luminosos codificados e é imune a interferências eletromagnéticas externas. Entretanto, são mais caras que os outros tipos de cabeamento, mas podem transmitir dados por grandes distâncias e em altíssimas taxas.

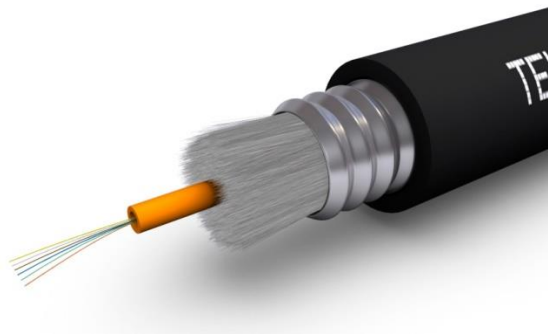


Figura 3.10 – Fibra óptica multimodo

Existem dois métodos de comunicação entre os nós de uma rede, mestre-escravo e produtor consumidor. Quando o método é mestre-escravo um nó, chamado de mestre, fica encarregado de gerenciar a transmissão na rede. Os demais nós só transmitem quando o mestre permitir de modo que o gerenciamento fica centralizado. A placa de comunicação SERCOS que interliga o CLP ao Driver Kinetix utiliza este método de comunicação.

No projeto atual foram utilizados cabos de fibra óptica para conectar os componentes que utilizam a rede DeviceNet e o Servo Drive ao cartão SERCOS (full-duplex), cabo RS232 para conectar o CLP ao computador e cabos UTP (Par trançado) para fazer a rede ethernet entre a câmera, o computador e o CLP. A rede DeviceNet utiliza a topologia de barramento, com todos os componentes ligados a uma linha de transmissão central que é conectada no módulo DeviceNet no blackplane do CLP.

### 3.3 Sercos

Trata-se de uma interface de comunicação em tempo real globalmente difundida, de licença aberta, para a comunicação entre dispositivos de movimento, dispositivos de entrada/saída e nós de internet.

O protocolo SERCOS (Real-time Communication System) define as camadas física, de enlace e de aplicação do padrão OSI. Na camada física, em topologia em anel, todas as informações são passadas apenas por um único laço de fibra óptica. A rede pode operar com taxas de transmissão de 2, 4, 8 e 16 Mbites/s e ciclos de 62.5, 125, 250, 500  $\mu$ s e múltiplos de 1ms, mas o controlador 1756-L60M03SE (da Allen Bradley) só opera com as taxas de 4 e 8 Mbites/s e ciclos de 0.5, 1 ou 2 ms.

A sincronia entre o mestre e seus escravos é mantida com o uso de mensagens regulares chamadas de telegramas. Todos os telegramas possuem 5 campos, sendo o primeiro e o último sequências delimitadoras contendo 01111110, um campo de endereço, um campo de dados e um campo para checagem de erros.

Para ordenar as transmissões no meio, a interface SERCOS define o *timing* de transmissão para cada nó da rede, de modo que dois escravos não responderão o mestre ao mesmo tempo. A sincronia do sistema é renovada a cada ciclo, de forma periódica, e envolve uma troca padrão de mensagens entre os escravos e o mestre.

Todas as mensagens possuem um identificador, chamado de IDN (*Identification Number*). O IDN está associado a uma função específica pré-definida na interface SERCOS e existem 32767 IDNs pré-definidos na especificação.

Para instalar o usuário deve conectar a saída TX (*Transmitter mode*) do cartão da interface SERCOS à entrada RX (*Receiver mode*) do primeiro nó (referente ao servo drive), a saída TX do drive ao RX do próximo nó, e assim sucessivamente, até que o laço de comunicação seja fechado com a conexão do último TX na entrada RX do cartão SERCOS.

O cartão de interface utilizado no projeto suporta até 3 eixos de movimentação. A configuração da rede foi feita utilizando o software RSLogix5000, da Rockwell Automation. Para o módulo utilizado, o 1756-L60M03SE, deve-se adicionar o módulo ao blackplane do CLP no software e inserir os dados de Baud rate de comunicação (4 ou 8 Mbites/s), adicionar os módulos do servo drive que foram conectados informando os nós onde eles se encontram e configurar os eixos.

### 3.4 DeviceNet

É um protocolo de rede, multi-drop para conexão entre sensores e atuadores (baixo nível) a controladores (alto nível), desenvolvido pela Allen-Bradley (atual Rockwell). A rede DeviceNet é classificada no nível chamado devicebus, cuja características principais são alta velocidade, alto poder de diagnóstico dos dispositivos da rede e comunicação a nível de byte englobando comunicação com equipamentos discretos e analógicos.

A rede foi utilizada para conectar os sensores indutivos, sendo dois sensores para fim de curso e três sensores para atuarem como botões de acionamento da bancada (liga, desliga e inicia rotina de deslocamento). Cada sensor recebe um endereço, sendo que o prefixo é igual para todos, mudando apenas o último bit. Vale ressaltar que estes sensores poderiam ser acoplados ao controlador utilizando um módulo de entrada digital, mas a

utilização com a rede DeviceNet é mais simples, rápida e utiliza menos fios, visto que todos os sensores são ligados no módulo (localizado próximo dos sensores) e este é conectado ao CLP por apenas um fio. O barramento utilizado foi o cartão DeviceNet modelo 1756-DNM visto na Figura 3.11.



Figura 3.11 – Cartão DeviceNet modelo 1756-DNM

### 3.5 Controlador Lógico Programável

Os Controladores Lógicos Programáveis (CLPs) nasceram na Hydronic Division da General Motors, dentro da indústria automobilística americana, em 1968. A necessidade de sua criação foi devido a grande dificuldade de mudar a lógica de controle dos painéis de comando (compostos por reles) a cada mudança na linha de montagem. Na indústria, de modo geral, cada pausa significa desperdício de tempo e altos gastos de dinheiro.

O projeto foi sob a liderança do engenheiro Richard Morley, que desenvolveu um equipamento versátil e de fácil utilização. Desde então, os controladores vem se aprimorando cada vez mais e mais, existindo cada vez mais modelos, variedades de entradas e saídas, maior velocidade de processamento, blocos lógicos complexos, funções de controle, interfaces sofisticadas e intuitivas, novas linguagens, protocolos, tipos de rede e aplicações.

Atualmente os CLPs estão na quinta geração, sua programação é feita pelo computador (em diversas linguagens diferentes), existem simuladores e, em alguns casos, debugadores para auxiliar na programação, há uma padronização dos protocolos de comunicação facilitando a integração de componentes e sistemas e os sistemas podem ser supervisionados em tempo real, através de sistemas supervisórios.

Em geral os CLPs de médio/grande porte são compostos por 7 componentes básicos: CPU (Central Processing Unit – responsável pelo processamento lógico, contendo um microprocessador), memória (armazena o programa e dados da “I/O ImageTable”), módulos de entrada e saída (para conexão dos botões, sensores e atuadores), blackplane(também chamado de chassis, responsável pela conexão do controlador com os demais módulos), software de programação, interface de comunicação e fonte de alimentação.

Os CLPs funcionam realizando um ciclo em loop, chamado ciclo de scan. O CLP, depois de inicializado, executa as seguintes atividades em loop: verifica o estado das entradas e atualiza a memória, executa o código e atualiza as saídas (conforme a Figura 3.12). Essas tarefas são executadas de forma simultânea ao processo e os status das entradas são copiados para uma região específica da memória chamada de “I/O ImageTable”.

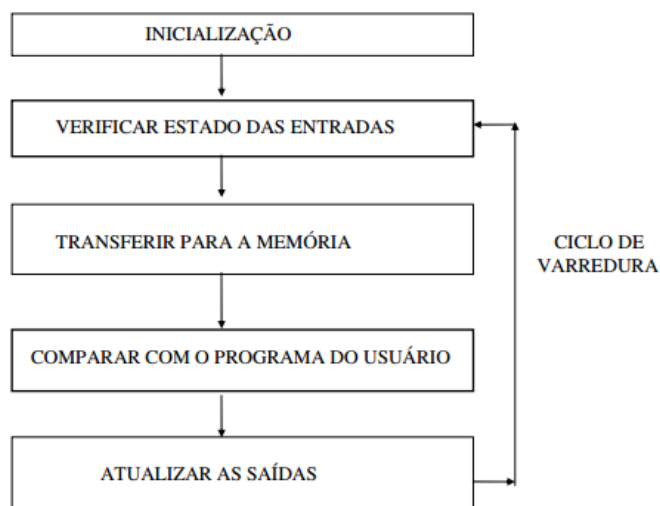


Figura 3.12 – Ciclo de scan do CLP

O programa é executado da primeira a última linha e atualiza a “ImageTable” com os status das novas saídas. A velocidade do ciclo de leitura e atualização depende do tamanho do programa e da capacidade de processamento do CLP.

Foi utilizado o CLP modular ControlLogix 5560, da Rockwell Automation, equipado com um rack com trilhos para alimentação e comunicação dos módulos e slots para o encaixe de outros módulos. Em cada rack se utiliza apenas um controlador e os outros são utilizados para módulo de entrada e saída (digitais e analógicos) e módulos de comunicação (como o DeviceNet).

No projeto foi utilizado um rack com 4 slots, sendo numerados de 0 a 3, acomodando um módulo de controle 1756-L60M03SE (Figura 3.13), que ocupa os slots 0 e 1 (pelo fato de possuir um módulo de comunicação SERCOS embutido), um módulo de comunicação DeviceNet 1756-DNM (Figura 3.11) no slot 2 e o slot 3 está livre.



Figura 3.13 – Controlador modelo 1756-L60M03SE

A programação da maioria dos controladores da Rockwell pode ser feita em SFC (Sequential Flow Chart), Ladder, Block Diagram e Structured Text. Independente da linguagem escolhida a arquitetura disponibiliza diversas funções comuns como timers, contadores e funções lógicas, além das funções específicas de cada linguagem para determinada aplicação. As instruções vão da lógica de contatos por relays, blocos de controle PID (Proporcional – Integral - Derivativo) até instruções para variadores de frequência integrados com o módulo SERCOS, como o Kinetix que foi utilizado no projeto. Além disso, é possível programar em várias linguagens ao mesmo tempo e, inclusive, chamar rotinas que tenham sido feitas em outra linguagem em seu programa principal.

O software RSLinx (Figura 3.14) é utilizado para estabelecer e gerenciar a rede de comunicação entre o controlador, seus módulos e o computador. Além disso, nele é configurado os módulos do backplane e, automaticamente, é feito o diagnóstico de cada componente, facilitando a manutenção. O RSLinx também permite a integração com outros programas, como o Factory Talk para terminais de acesso remoto para operadores, RSNetWorx para configuração da rede DeviceNet, dentre outros.



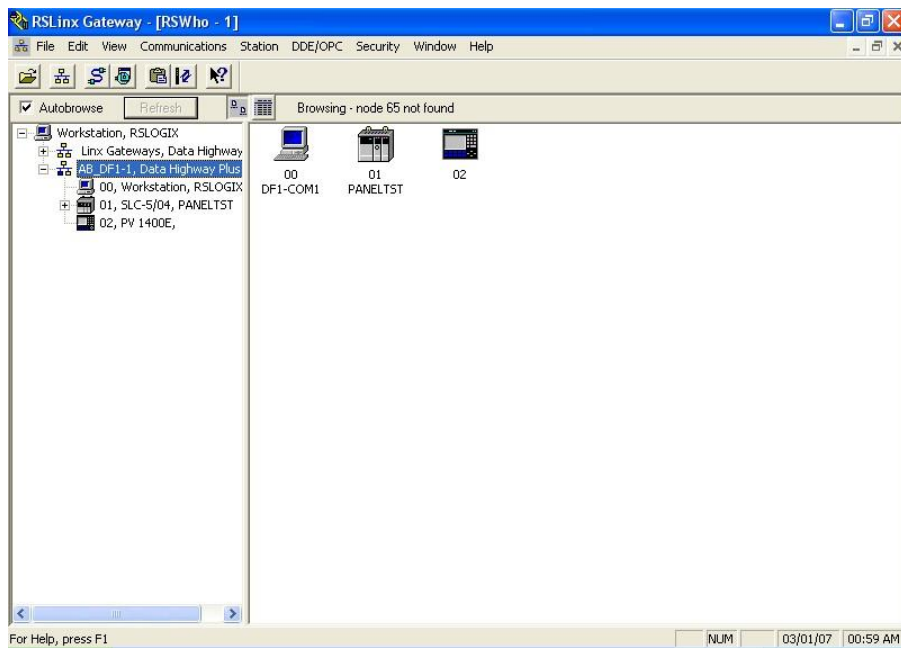


Figura 3.14 – Interface do software RSLogix

O software utilizado para a programação foi o RSLogix5000 (Figura 3.15). Nele é possível desenvolver programas simbólicos nas linguagens supracitadas, criando variáveis lógicas e reais, de entrada e saída, discretas e contínuas, além de arrays, estruturas e rotinas. Durante a execução é possível acessar os valores das variáveis em “tempo real”, com um pequeno atraso. As variáveis são divididas em ControllerTags (variáveis reais de controle) e ProgramTags (variáveis lógicas criadas para auxiliar na programação).

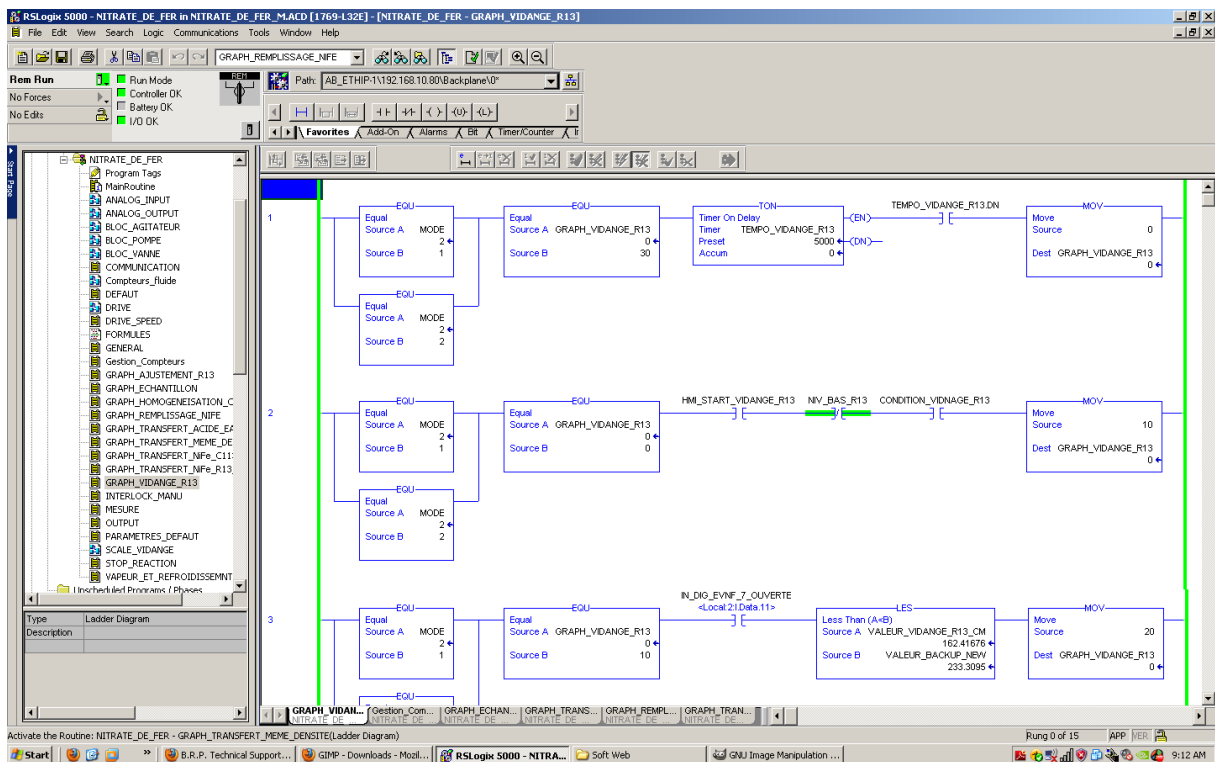


Figura 3.15 – Interface do software RSLogix5000

Os dois tipos de variáveis podem ser visualizadas em tempo real por uma ferramenta chamada Trend (Figura 3.16). Ela mede as variáveis e pode construir gráficos com os valores de cada uma à medida que o tempo passa. É possível, por exemplo, monitorar a posição, velocidade e aceleração do servo motor.



Figura 3.16 – Ferramenta Trend do RSLogix5000

No projeto o CLP é utilizado para controlar o servo motor (MPL-A310F-SJ22AA), através do drive Kinetix (2094-AC05MP5), de modo que o carro da ponte rolante se desloque seguindo a trajetória planejada. O drive recebe *feedback* do encoder do servo motor de modo a aumentar a precisão durante o deslocamento, possuindo uma malha de controle para a posição e velocidade do giro do motor. O CLP comunica com o drive através de um par de fibras ópticas full-duplex, utilizando a interface SERCOS. O cartão 1756-DNB (da rede DeviceNet) está configurado para receber informações dos sensores indutivos (modelo 871T-R8B18). Os dois sensores configurados como sensores fim de curso são responsáveis por evitar que o carro saia dos limites físicos da ponte rolante. Quando um deles é acionado o CLP entra em uma rotina de emergência e desliga o servo motor, causando uma parada brusca.

### 3.6 Line interface module 2094-AL09

Este módulo é responsável por realizar a interface da rede trifásica com o módulo servo drive, para acionamento do servo motor. Este modelo (Figura 3.17) possui três conjuntos de disjuntores [15], sendo CB1 (liga ou desliga o fornecimento de tensão trifásica ao drive), CB2 (fornece tensão monofásica para o servo drive) e CB3 (liga as duas fontes de tensão DC de 24V, responsáveis pela entradas e saídas digitais do módulo e alimentação do freio do motor).

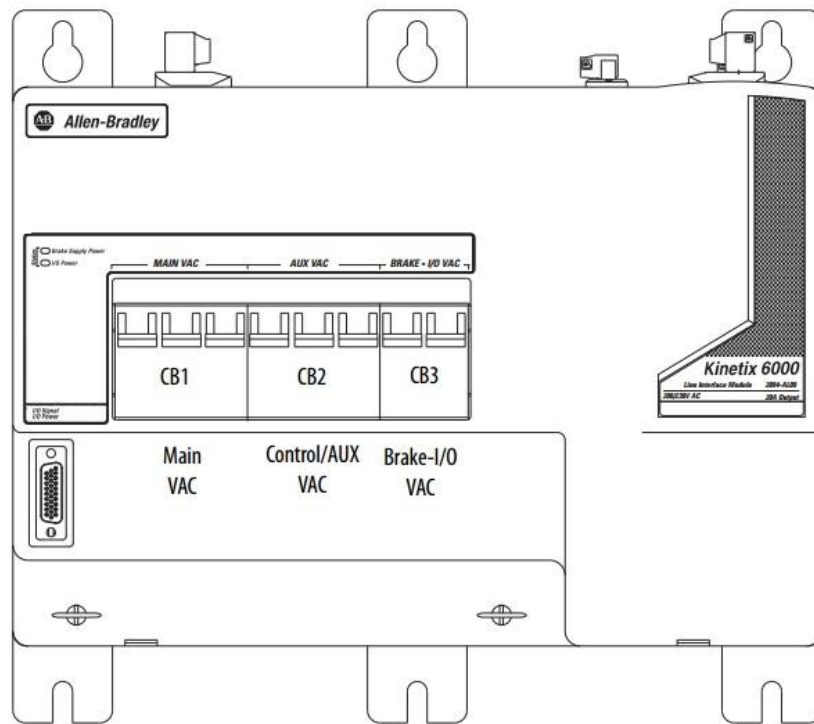


Figura 3.17 – Line interface module modelo 2094-AL09 da Allen Bradley

### 3.7 Servo drive

Sua função é o acionamento do servo motor. O driver fornece potência ao servo motor baseado no comando que vem do CLP. O controle da velocidade do servo motor é feito a partir da modularização do sinal fornecido, através de pulsos PWM (*pulse width modulation*). O Driver utilizado é o Kinetix 6000 (Figura 3.18), da Allen Bradley.



Figura 3.18 – Driver Kinetix 6000 da Allen Bradley

As especificações técnicas do driver são mostradas na Tabela 3.

Tabela 3 – Especificações técnicas do Kinetix 6000

Plataformas de controle	Compactlogix, ControlLogix e RSLogix 5000
Corrente de saída	3.7 a 34A rms
Potência de saída	1.2 a 22 KW
Tensão de entrada (trifásica)	195 a 265Vac, 324 a 528Vac
Realimentação	encoder (multivolta, uma volta, incremental, seno/coseno e resolver)
Função de segurança	Categ. 3 e SIL3
Compatibilidade de motores	Série F, MP, TL e 1326AB

O circuito que realiza a variação de frequência é similar ao mostrado na Figura 3.19. O circuito retificador transforma a corrente alternada na entrada em corrente contínua e os capacitores tem a função de diminuir o efeito de *ripple*. Os transistores são acionados por um sinal PWM para realizar o chaveamento, mas o circuito que gera este sinal de chaveamento não está disponível na literatura disponível do equipamento 2094-AC05MP5.

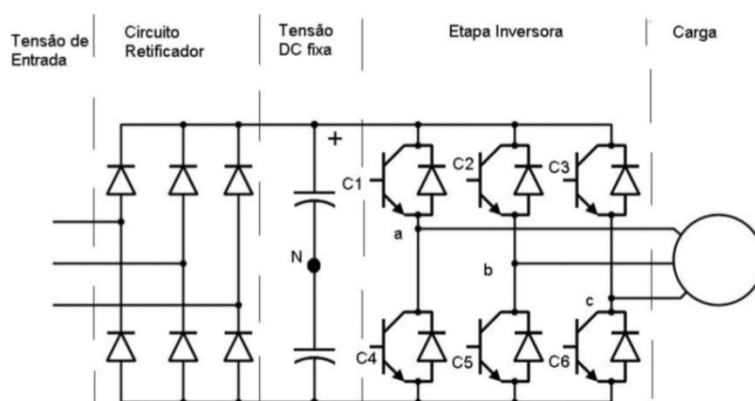


Figura 3.19 - Circuito para gerar o sinal PWM

Deseja-se uma saída senoidal, logo o pulso do PWM varia conforme a amplitude do sinal senoidal causando uma réplica do PWM na saída, sendo que o valor médio do PWM será equivalente ao sinal senoidal, conforme a Figura 3.20.

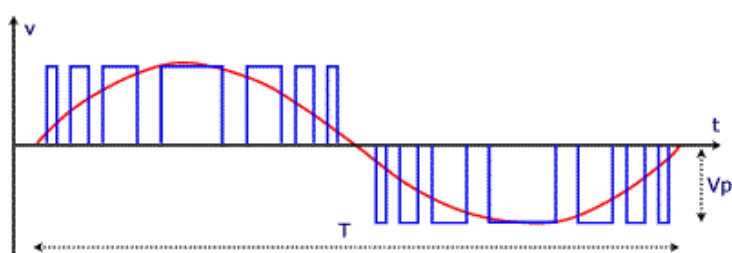


Figura 3.20 –Sinal PWM (em azul) e valor médio do sinal (em vermelho)

Quando o servo motor se desloca, o movimento é lido pelo encoder e transmitido para o servo drive, o qual encaminha os dados para o controlador. No controlador é feita a comparação dos dados lidos com o que foi enviado. A diferença entre essas duas grandezas gera um sinal de erro que é enviado ao drive, que ajusta a potência fornecida ao motor através do PWM. Isso se repete até que a diferença seja nula. Além das saídas de potência, o modelo utilizado possui três conjuntos de entradas na frente do módulo. Um conjunto de entradas digitais para sensores e botões de comando, um conjunto de entradas próprias para ligação do sinal de *feedback* do motor e um conjunto de entradas para sinal de um encoder extra. Utilizou-se apenas a entrada de *feedback* do motor, destinada ao encoder do tipo Stegmann Hiperface, o qual faz parte do próprio servo motor (MPL-A310F-SJ22AA), e aceita entradas seno, cosseno e de flags do motor, além de possuir uma fonte de 5 ou 9 V para a alimentação do encoder. O circuito interno completo do servo drive pode ser visto na Figura 3.21.

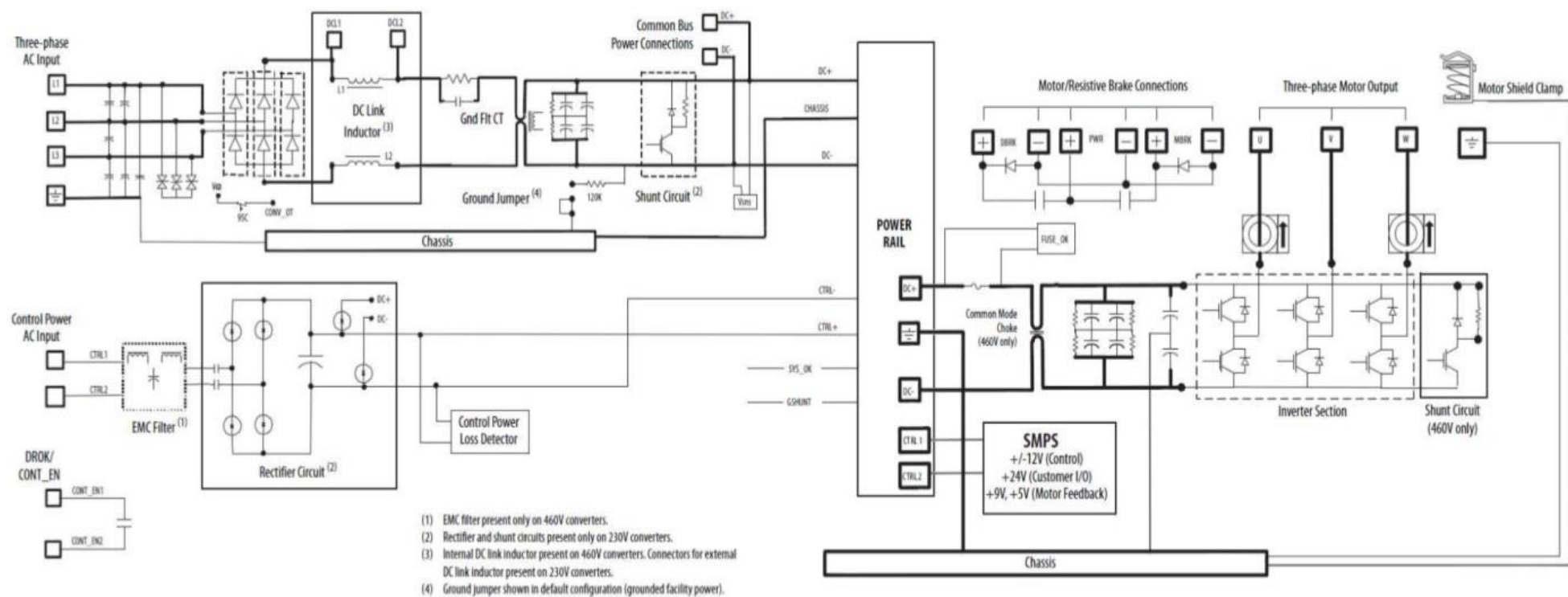


Figura 3.21 – Circuito interno do servo drive [18].

### 3.8 Servo Motor

Os servo motores são usados em várias aplicações quando se deseja movimentar algo de forma precisa e controlada. É composto por um motor acoplado a um sensor (encoder) que mede sua posição e velocidade, de modo a possibilitar o *feedback* do seu estado para o sistema que o controla, gerando assim um sinal de erro, afim de que o sistema de controle altere o acionamento do motor para minimizá-lo. Quando a velocidade e a posição se aproximam dos valores de referência, o sinal de erro e, conseqüentemente, o sinal de controle se aproximam de zero. Em geral, os servo motores necessitam de um módulo específico para realizar o seu controle, como o driver Kinetix utilizado no projeto.

O modelo utilizado foi o servomotor MPL-A310F-SJ22AA da Allen Bradley, ilustrado na Figura 3.22. É composto por um motor indutivo, que opera a 230 V e um encoder do tipo Stegman Hiperface, que opera de forma absoluta e incremental, sendo que a posição é medida de forma absoluta e a velocidade doé medida de forma incremental. A parte absoluta possui código Gray de alta resolução, possibilitando uma leitura bastante precisa de posição.



Figura 3.22 – Servo motor modelo MPL-A310F-SJ22AA

### 3.9 Sensores Indutivos

São utilizados para identificar objetos metálicos a partir do princípio da variação do campo magnético causada por esses objetos, já que os sensores possuem uma bobina de indução eletricamente excitada.

Estes sensores, em geral, são alimentados por corrente contínua. Entretanto, para criar o campo magnético é necessária corrente alternada. Para possibilitar isto eles possuem um oscilador, que converte corrente contínua em corrente alternada, de forma senoidal. A maioria destes sensores possui um microprocessador, para condicionar o sinal antes de enviá-lo e se comunicar de acordo com a rede utilizada.

O modelo utilizado foi o sensor indutivo 871T-R8B18 (Figura 3.23). Ele possui duas bobinas, uma usada como referência e a outra para detectar as correntes magnéticas induzidas no objeto condutor. Este sensor é dito inteligente, pois possui processamento local próprio para se comunicar em uma rede DeviceNet. Este sensor pode ser utilizado para medir distâncias, com detecção nominal de 8mm. Entretanto, ele foi utilizado em uma configuração discreta indicando apenas se há um objeto metálico a sua frente. Ele possui bits para identificação do objeto, a proximidade e bits de autodiagnóstico, para evitar leituras provenientes de um sensor defeituoso.



Figura 3.23 - Sensor indutivo 871T-R8B18

### 3.10 Encoders

Os encoders são dispositivos utilizados para medir rotação e velocidade angular. Existem dois tipos de encoders, os incrementais e os absolutos, podendo alguns modelos possuir as duas codificações, como o que foi utilizado (Figura 3.24). Os encoders incrementais só podem determinar sua posição angular com base em um referencial que geralmente é uma marcação única em uma das trilhas internas do disco. Alguns encoders incrementais são utilizados com tratamento dos dados por software, considerando a posição inicial como a referência. Neste projeto o sensor responsável por medir a velocidade instantânea do servo é do tipo incremental. Já os encoders absolutos possuem uma marcação diferente para cada posição angular. Dois tipos de codificação são muito utilizadas nestes encoders, o código binário e o código gray. Para medir a posição angular do servo motor utilizou-se um encoder absoluto, codificado em código gray.



Figura 3.24 – Exemplo de encoder incremental e absoluto



### 3.11 Câmera PresencePlus

A câmera utilizada foi a PresencePlus P4 GEO, da Banner Engineering (Figura 3.25). Esta câmera foi desenvolvida para uso em ambiente industrial, sendo um sensor robusto e fácil de utilizar, para aplicações de inspeção automatizada através de visão computacional. Possui resolução ajustável de até 1.3 megapixels, com acionamento através de *Trigger*, de forma que a taxa de amostragem pode ser ajustada de acordo com a aplicação [16].



Figura 3.25 – Câmera utilizada no projeto, da Banner Engineering

Utiliza a ferramenta GEO Count para detectar presença, localização e rotação do padrão do alvo. É possível inspecionar características dos objetos independente da orientação e da posição do mesmo no campo de visão.

Além disso, a câmera vem com o PPROCT 1.3, que é o controlador de sistema de visão (Figura 3.26). O processamento de imagens pode ser feito nesta placa, além de ela conter todos os I/O, alimentação e saídas para comunicação em rede.



Figura 3.26 - Placa de processamento da câmera

### 3.12 Programa Ladder

A programação do CLP poderia ser feita em qualquer linguagem suportada pelo controlador da Rockwell. Entretanto, o servo motor Kinetix possui um conjunto de instruções em Ladder específicas para o seu acionamento, chamadas de *motion group* [14]. Por este motivo, toda a programação foi feita em Ladder.

Ladder, do inglês escada, foi a primeira programação criada para uso nos controladores lógicos programáveis. Formada por duas linhas verticais, sendo uma com tensão e uma aterrada. Outras linhas conectam as duas, de forma horizontal, com entradas e saídas, ficando semelhante a uma escada (Figura 3.27).

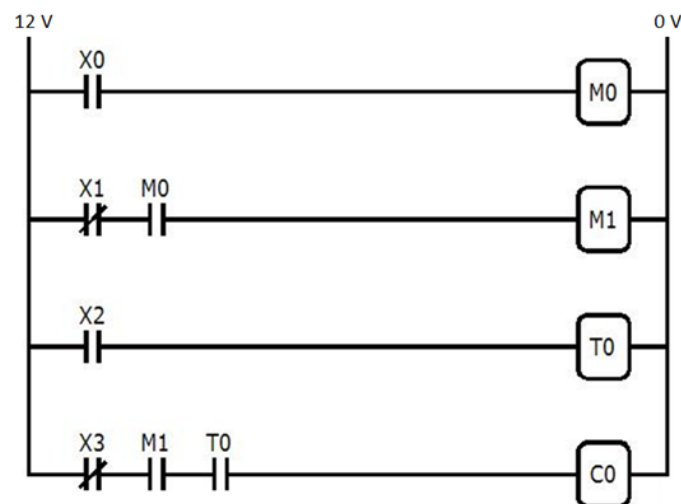


Figura 3.27 – Exemplo de programa em Ladder

Esta linguagem visual é simples de ser utilizada e procura imitar os antigos diagramas elétricos e contatos por relés permitindo, na época, uma maior facilidade de transição e aceitação dos funcionários. Para exemplificar, a Figura 3.28 mostra um diagrama elétrico de uma partida direta e a Figura 3.29 o Ladder equivalente.

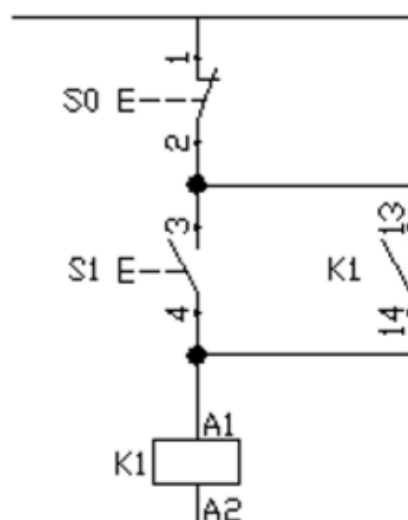


Figura 3.28 – Diagrama elétrico de uma partida direta



Figura 3.29 – Ladder para partida direta

Para que o servo realize o movimento desejado é necessário, inicialmente, utilizar as instruções MSO e MAH, de ativação do eixo (Figuras 3.30 e 3.31). Depois é possível alterar a dinâmica do servo com a instrução MCD (Figura 3.32)

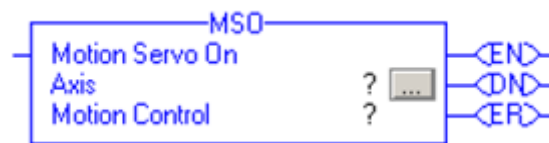


Figura 3.30 - Motion Servo On – ligar o servo

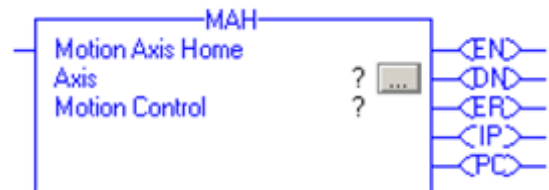


Figura 3.31 - Motion Axis Home – utilizado uma única vez, no início da movimentação, para selecionar o ponto inicial

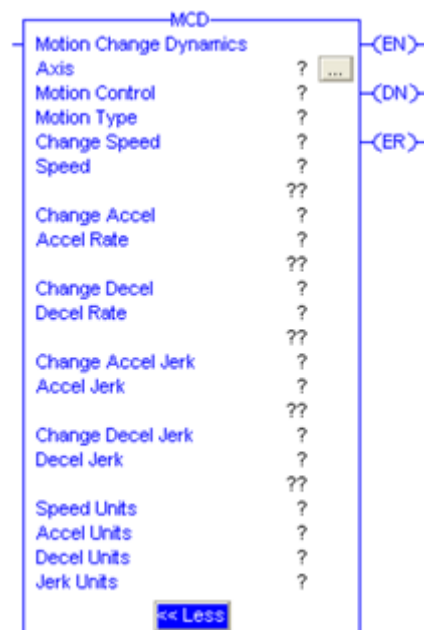


Figura 3.32 - Motion Change Dynamics – atua diretamente sobre o servo, realizando a sua movimentação de acordo com os parâmetros escolhidos

O motor utilizado não possui a opção de trabalhar em malha aberta, trabalhando em malha fechada com o drive Kinetix. Dessa forma, não foi possível realizar ensaios para modelar a dinâmica do motor em relação a um sinal de tensão ou corrente em sua entrada. O servo motor possui realimentação tanto para sua velocidade quanto para sua posição, de modo que as configurações podem ser alteradas através do RSLogix 5000, no menu *AxisProperties* – *main\_axis*. Os diagramas das malhas de controle de posição e velocidade podem ser encontrados no manual [20], apêndix D, *AXIS\_SERVO*.

Para calibrar os ganhos do motor, inicialmente, foi utilizada a opção de *auto-tuning*, com os dados do motor utilizado e a segunda opção de calibração – *tracking applications* (aplicações de rastreamento), conforme o manual [8]. O *auto-tuning* selecionou valores para um ganho proporcional para a malha de velocidade e os ganhos proporcional e integral para a malha de posição.

Alguns testes foram feitos na bancada com os ganhos do *auto-tune*, sem nenhum carregamento no carro da ponte. A posição de referência era atingida com precisão e rapidez, já a velocidade possuía um grande erro durante o rastreamento em relação a uma entrada rampa, com aceleração de 100 voltas/s<sup>2</sup> para atingir uma velocidade final de 10 voltas/s.

Para corrigir este erro foi escolhido um ganho integral para a malha de velocidade. À medida que eram feitos testes na bancada, observando o Trend entre a referência e a posição/velocidade atingida pelo servo motor, o ganho foi sendo calibrado, de forma empírica. O valor final dos ganhos pode ser observado na Figura 3.33 e as outras configurações na Figura 3.34.

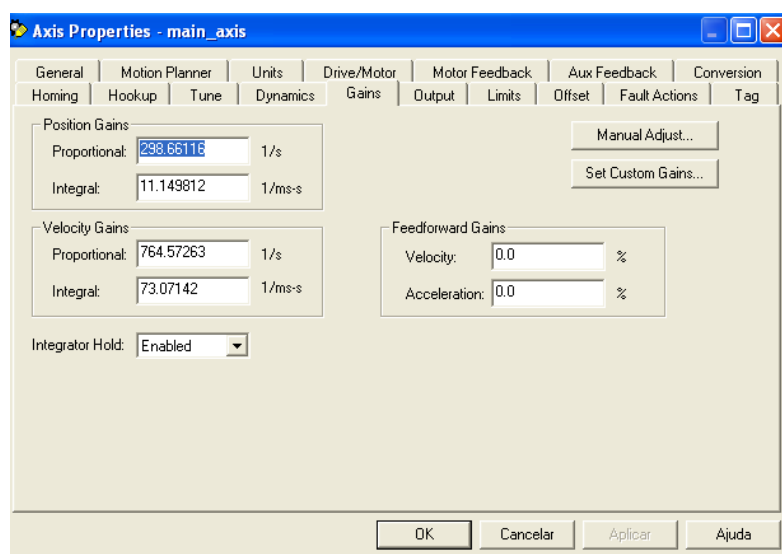


Figura 3.33 – Configuração dos ganhos proporcional e integral no driver

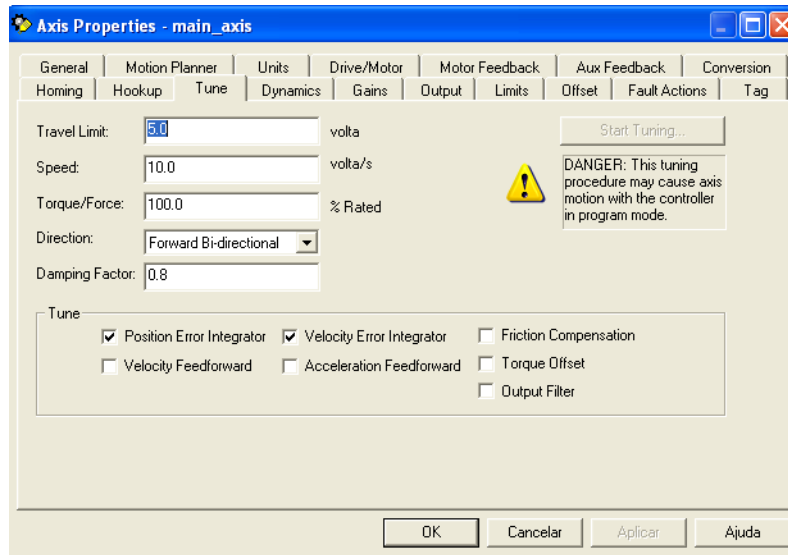


Figura 3.34 – Parâmetros limite para o servo

O Manual aconselha a utilização do ganho integral apenas na componente mais importante para a aplicação, sendo a posição ou a velocidade, informando que altos ganhos em ambos podem comprometer o desempenho do sistema. Entretanto, o ganho integral gerado pelo *auto-tune* para a malha de posição é bem inferior ao ganho escolhido para a malha de velocidade e os resultados foram satisfatórios.

Com os valores utilizados, o motor consegue seguir uma referência com inclinação de 100 voltas/s<sup>2</sup> com um erro de atraso muito pequeno (0.012 segundos). Como nossa aplicação trabalha com acelerações muito inferiores (cerca de 15 voltas/s<sup>2</sup>), é considerado que o motor seguira fielmente a entrada imposta.

Além disso, é possível selecionar o torque que o motor irá trabalhar (escala percentual) e sua velocidade limite de modo a proteger o servomotor durante o seu uso. O manual recomenda ainda a não utilizar o *feedforward* de velocidade e de aceleração nos primeiros testes e só incluir caso o desempenho desejado não seja atingido. Na aplicação eles não foram utilizados.

A figura 2.1, mostrada no Capítulo 2, ilustra a posição que o carro deve assumir para que a ponta cumpra a trajetória desejada, de forma suave. Para que o CLP controle o motor durante esta trajetória é necessário que ele passe diversos pontos de referência para o drive do motor. Entretanto, ao utilizar uma série de instruções com as posições em que o carro deve se posicionar, o tempo para estabilizar o servo nestas posições ultrapassa o tempo para que ele chegue até elas, de modo que o movimento é feito com leves trepidações, o que é suficiente para comprometer o experimento. Este problema foi observado inclusive com o uso do CAM Profile, conjunto de instruções específicas para controle de posição de servo motores através do driver.

Porém, foi observado que a instrução que alterar a velocidade do carro (MCD) é executada de forma “instantânea”, possibilitando o cumprimento da trajetória sem pausas no meio do caminho. O próximo passo foi a criação de um perfil de velocidade a partir do perfil de posição adquirido.

### 3.13 Criando o perfil de velocidade

O perfil de posição criado é composto de cerca de 400 pontos, podendo variar de acordo com a distância selecionada para o deslocamento. Outro detalhe, além de transformar o perfil de posição em um perfil de velocidade, é a quantidade excessiva de pontos que torna o código do CLP grande e, conseqüentemente, lento.

Para gerar a velocidade (derivada temporal da posição) e reduzir o número de pontos da trajetória foram utilizados os dois passos a seguir:

1º. Cálculo das derivadas discretas (pela definição), de dois em dois pontos, gerando um gráfico de velocidade (Figura 3.35);

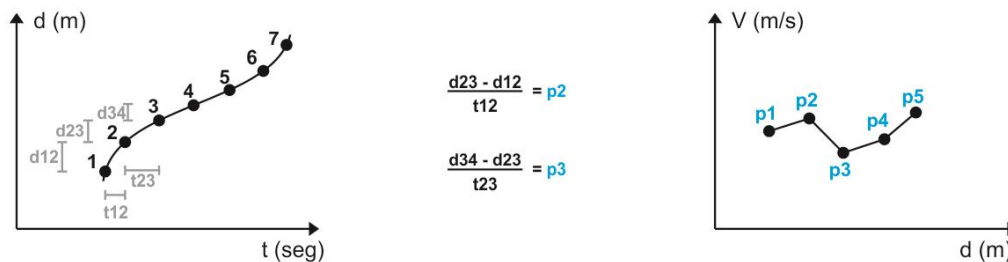


Figura 3.35 – Gerando a derivada da trajetória de posição

2º. Cálculo da média entre os pontos de velocidade, gerando um novo gráfico de velocidade média, com um número reduzido de pontos (Figura 3.36).

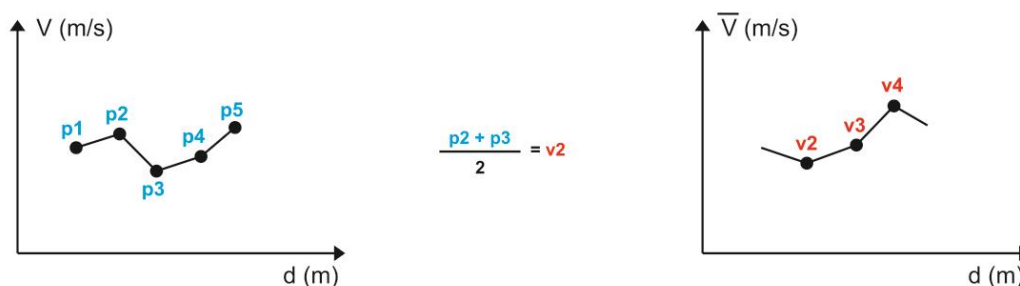


Figura 3.36 – Diminuindo a quantidade de pontos da trajetória

Nos gráficos de velocidade, no eixo das abcissas foi utilizada a distância percorrida, para obter um gráfico de velocidade x posição, para facilitar a inserção do comando em Ladder no CLP. No segundo passo, responsável por diminuir a quantidade de pontos, foram testados a média entre 2, 3, 4 e 5 pontos, a fim de diminuir consideravelmente o número de pontos e o tempo para implementação no CLP. A diferença encontrada entre os perfis de

velocidade foi mínima e, após alguns testes, foi definido o uso de 4 pontos para o cálculo da velocidade média.

A Figura 3.37 mostra a trajetória de velocidade criada a partir do perfil de posição ilustrado na Figura 2.1.

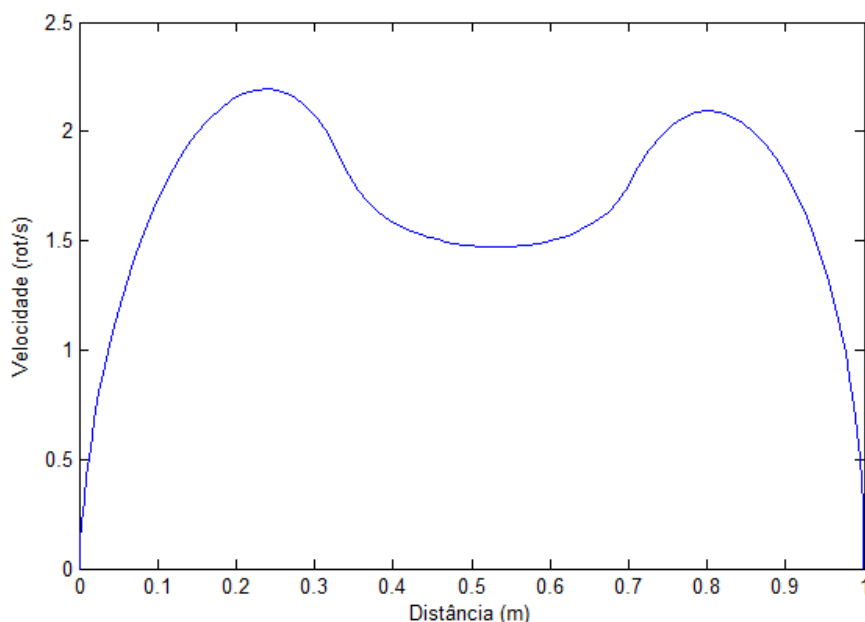


Figura 3.37 – Perfil de velocidade

Para facilitar a programação do CLP, a velocidade foi convertida para rotações/segundo, utilizando um fator de conversão de 0.072, já que uma rotação do servo motor equivale a 0.072 metros.

Além disso, durante a obtenção da derivada é calculado o tempo médio para fazer a trajetória utilizando o perfil de velocidade. Esse tempo é utilizado para construir um perfil de velocidade constante para comparar os resultados (simulando algo próximo ao deslocamento do operador da plataforma), de modo que as duas movimentações (perfil de controle e velocidade constante) demorem o mesmo tempo para concluir o movimento.

Agora temos a velocidade que o motor deve estar em uma determinada posição. Com o uso da instrução MCD podemos impor uma determinada velocidade. Com o uso do encoder do servo podemos medir a posição exata do carro. Logo, o código em Ladder para a trajetória se torna simples, de modo que ele deve ler a posição e, para cada ponto pré-estabelecido, alterar a velocidade do motor. O código de execução mais rápida ficou semelhante a uma máquina de estados, onde as entradas são a velocidade e a posição atual do carro e a saída é a nova velocidade.

O programa do CLP foi dividido em 5 rotinas, de forma a ter um programa modulado e simples de programar e corrigir erros. Uma rotina de inicialização da rede DeviceNet e uma de inicialização do eixo do servo, utilizando as instruções Motion Servo On (MSO), Motion Axis Stop (MAS) e Motion Axis Home (MAH), para zerar a configuração de deslocamento. Uma rotina de segurança, ativada pelas flags de erro do driver e pelos

sensores fim de curso. Nesta rotina a energia do servo motor é cortada e seu eixo é travado, com o uso da instrução Motion Servo Off (MSF). As outras duas rotinas são utilizadas para escolher o perfil de velocidade (através dos sensores em configuração de botões) e armazenar a trajetória, com os dados de velocidade e posição que o servo deve assumir.

O programa está sempre executando a rotina principal (que possui a trajetória) e a partir de algum evento, como a leitura do sensor fim de curso, o programa ativa a rotina correspondente, como a de segurança, por exemplo.

O perfil de velocidade média criado, mostrado na Figura 3.37, foi implementado e executado na planta. O resultado da posição do carro durante o movimento pode ser visualizado na Figura 3.38, com o uso da ferramenta Trend.

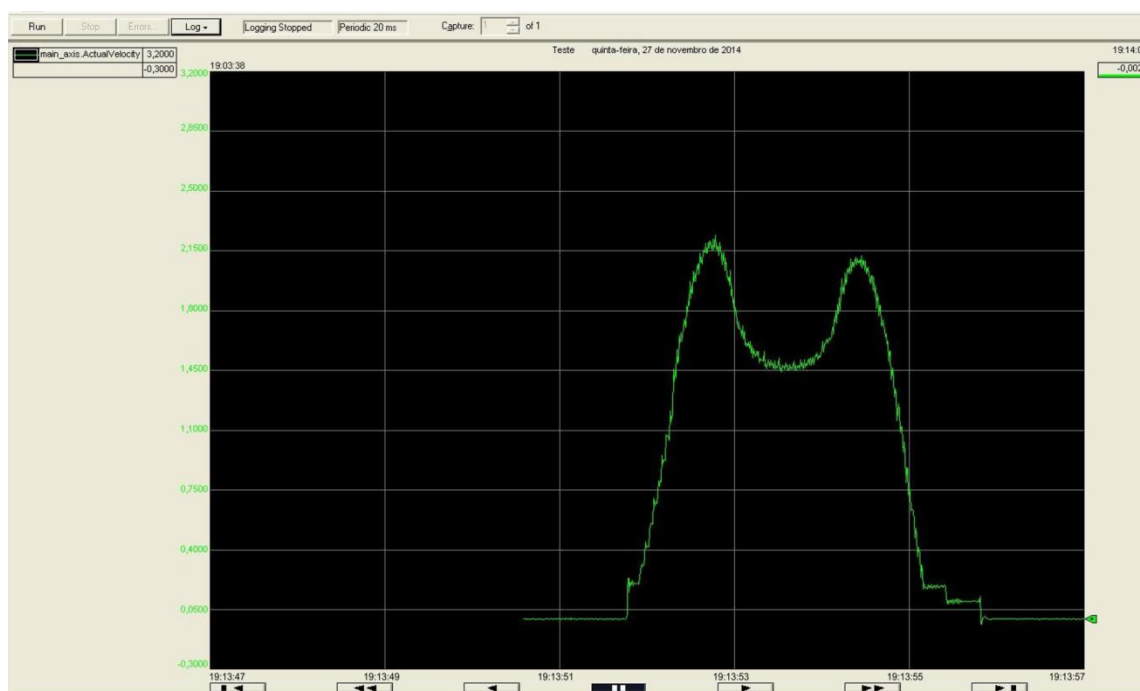


Figura 3.38 – Leitura no Trend para o perfil de velocidade

Este perfil medido pode ser comparado com o teórico gerado no Matlab, a fim de quantizar o erro entre a trajetória imposta e a realizada pelo carro. No início e no final do movimento podemos observar variações bruscas na velocidade. Entretanto, após alguns testes, concluiu-se que o erro é devido ao a imprecisão em baixas velocidades e ao método de cálculo da velocidade a partir dos dados do encoder. Antes de se locomover o carro está com velocidade nula. O cálculo da velocidade instantânea, realizada pelo driver, é feito através da coleta de uma série de pontos em pequenos intervalos de tempo, de modo que no início e no final do movimento a velocidade nula entra no cálculo.

Utilizando a ferramenta de rastreamento de bitmap, do software CorelDRAW, é possível sobrepor o gráfico de velocidade teórico, gerado pelo Matlab, e o gráfico de posição medida, gerado através da medição do encoder. É possível observar a sobreposição na Figura 3.39.



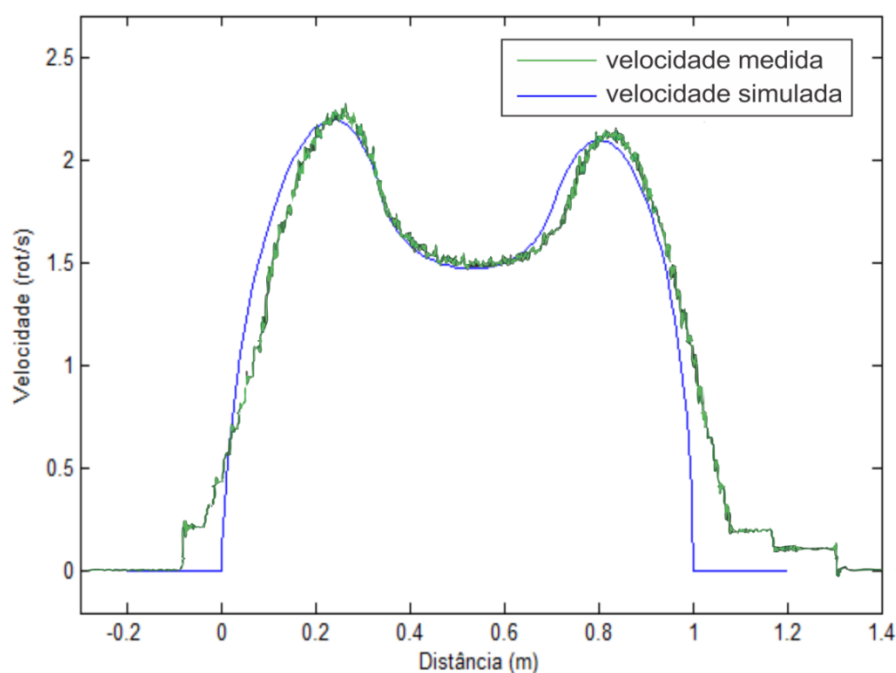


Figura 3.39 – Sobreposição da velocidade teórica e da medida com o encoder

A princípio a disparidade entre as duas curvas é grande. Para analisar melhor foi gerado um gráfico com o deslocamento da velocidade medida para a esquerda, de forma a alinhar os dois pontos de máximo, como ilustra a Figura 3.40.

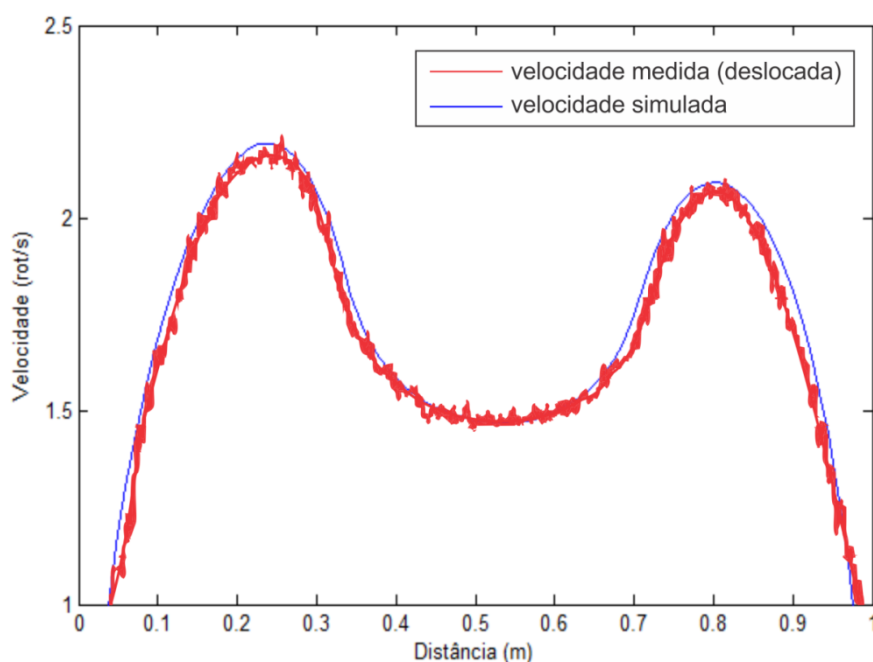


Figura 3.40 – Sobreposição com a curva da velocidade medida deslocada

Desta forma é possível observar que durante grande parte do deslocamento a curva é real é bem próxima da curva teórica. Apenas no início e no final do perfil de velocidade que o motor não consegue ser preciso ou, como comentado anteriormente, o encoder não consegue medir com precisão suficiente.

# AQUISIÇÃO E PROCESSAMENTO DE IMAGENS, ANÁLISE E CONCLUSÃO

### 4.1 Introdução

Apesar de a cena real ser uma entidade tridimensional, uma imagem pode ser representada por uma função bidimensional  $F(x,y)$ , onde  $x$  e  $y$  são variáveis espaciais.  $F(x,y)$  pode ser caracterizada a partir da quantidade de luminosidade incidente (iluminação),  $i(x,y)$ , e da luz refletida pelos objetos que a compõem (reflectância),  $r(x,y)$ .

$$f(x,y) = i(x,y) r(x,y) \quad 24)$$

Uma imagem pode ser definida de forma monocromática (preto e branco) e chama-se *nível de cinza* a sua intensidade e a escala formada por todos os possíveis valores é chamada de escala de cinza. A Figura 4.1, a seguir, ilustra a imagem da bancada em tons de cinza 8 bits.

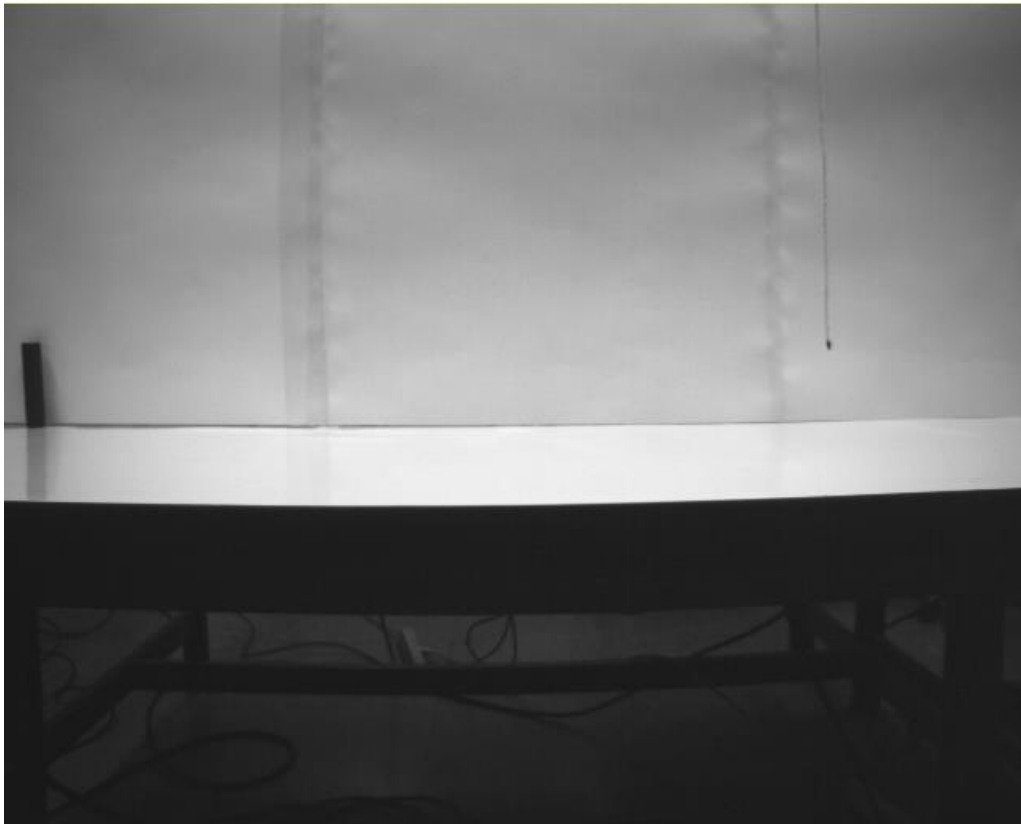


Figura 4.1 – Bancada em tons de cinza 8 bits

A amostragem da imagem consiste na discretização do domínio da função  $f(x,y)$ , transformando-o em uma grade de pontos regularmente espaçados entre si, de forma análoga a uma matriz. Já a quantização da imagem define um conjunto finito de níveis de cinza que irão representar a imagem.

Além disso, podemos representar a imagem em cores utilizando a escala RGB (do inglês *red, green and blue*) que, com a junção dos níveis das cores vermelho, verde e azul, podem formar qualquer imagem colorida.

A câmera utilizada não capta cores, entretanto para nossa aplicação a imagem pode ser utilizada em escala de cinza, desde que o contraste entre o objeto a ser identificado e o fundo seja suficiente. Para aumentar o contraste e, conseqüentemente, a precisão da identificação, foi utilizada cartolina de cor branca, fixada no fundo da bancada, e identificador na ponta do barbante com a coloração preta.

## 4.2 Calibração da câmera

A câmera foi fixada a uma distância de cerca de 120 centímetros da bancada e a uma altura de 110 centímetros. Para focar na área de movimentação útil foi utilizado o zoom da câmera em 2x, como ilustrado na Figura 4.2.

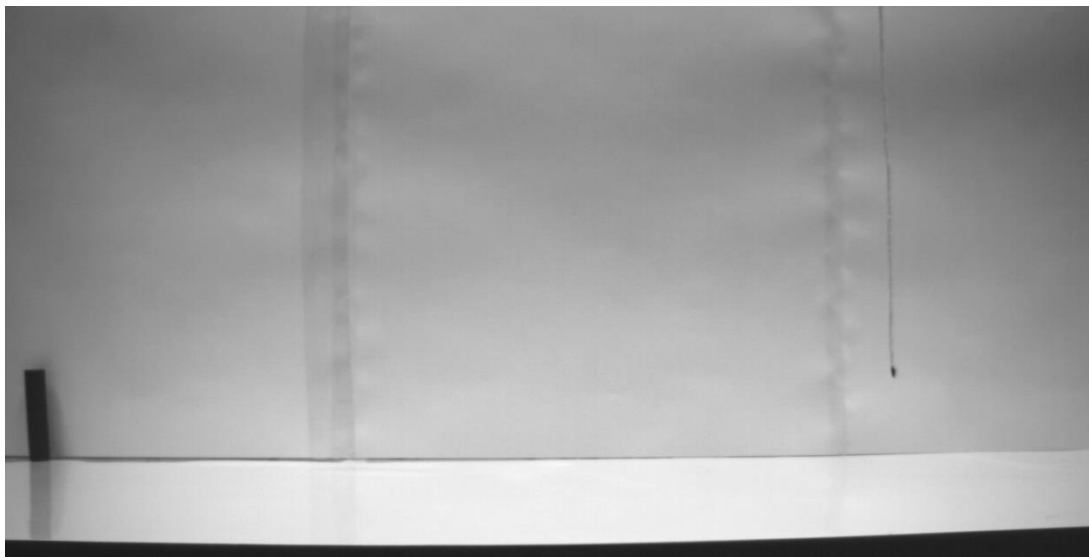


Figura 4.2 – Imagem da bancada com o zoom utilizado

A barra preta no canto inferior a esquerda foi utilizada como ponto fixo na imagem, para se ter um sistema de coordenadas de referência. Para melhorar ainda mais o contraste na área de interesse, foi alterado o tempo de exposição da câmera de 10.88 ms (*default*) para 23.20 ms, como se segue na Figura 4.3.



Figura 4.3 – Imagem da bancada com o contraste utilizado

Desta forma o contraste é muito maior entre a ponta do barbante e o fundo. Além disso, as sombras e os pequenos tons de cinza que estavam no caminho a ser percorrido sumiram.

Quando obtivermos os dados da câmera não teremos uma posição exata, mas sim uma localização em relação aos pixels. Para converter os pixels em uma posição foi feita uma calibração dos dados recebidos de duas formas. Primeiramente, foi afixada uma régua na bancada para medir a relação entre espaço e pixels, de modo que a relação média foi de 12.2 pixels/cm. Posteriormente, foi feita a relação entre uma movimentação conhecida do carro, utilizando o encoder, e observada a relação entre os pixels capturados pela câmera. A relação média foi de 11.7 pixels/cm. O valor utilizado foi o adquirido através da calibração do encoder, visto que este dado será utilizado como entrada para o sistema que controla a velocidade do servo motor. Foi observada uma pequena diferença entre a distância equivalente aos pixels nas extremidades da imagem e no centro, próximos ao foco da câmera. Entretanto, a diferença é consideravelmente pequena e foi desprezada. Esta câmera possui também um fator de correção para minimizar o erro de paralaxe.

### **4.3 Aquisição e processamento das imagens**

Após a calibração da câmera, podemos aplicar os filtros e a técnica de identificação desejada. A implementação poderia ser feita utilizando diversas ferramentas de visão computacional e diversas linguagens para o processamento de imagens, como Python, Matlab, java, C/C++ com o uso de OpenCV (*Open Source Computer Vision Library*), sendo cada uma adequada para uma determinada aplicação.

Neste trabalho isto foi feito de duas formas, utilizando o Matlab e o processamento da própria câmera. Primeiramente, foi feito com a aquisição das imagens através da câmera e o seu processamento no Matlab. Este software possui diversas funções da área de visão

computacional implementadas e é consideravelmente fácil o manuseio das matrizes (percorrer, alterar, realizar operações matemáticas) no ambiente do software. Entretanto, deseja-se desenvolver um sistema que, futuramente, possa funcionar em tempo real, o que seria inviável utilizando o processamento no Matlab, devido à necessidade de uso de vários filtros e da grande quantidade de varreduras da matriz para cálculo do centro geométrico dos pixels da cor que se deseja rastrear.

A segunda forma foi utilizar o pacote de ferramentas de visão computacional da Banner, contido no software PresencePlus.

A princípio são selecionadas as ferramentas (*tools*) a serem utilizadas. No nosso caso foram utilizadas as ferramentas de localização e de padrão geométrico (Figura 4.4).

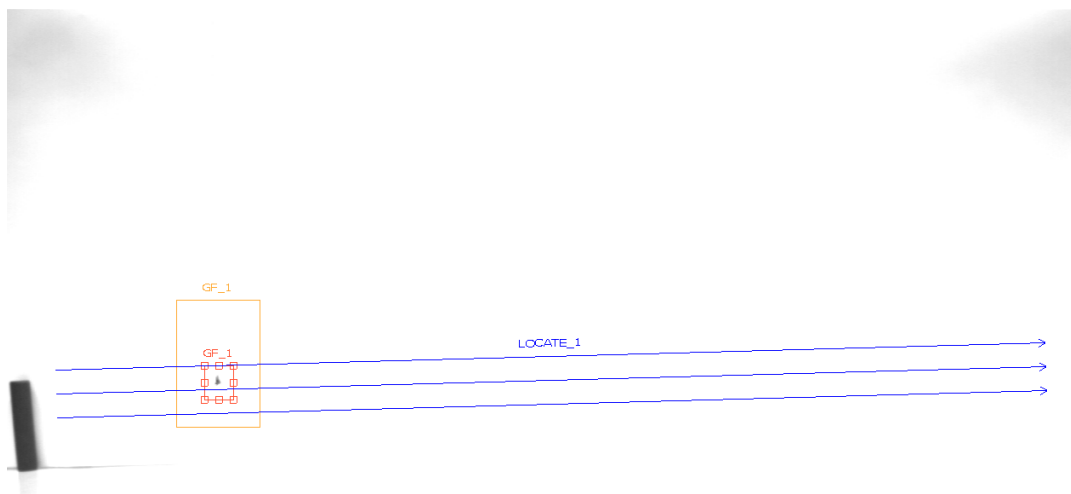


Figura 4.4 – Exemplo da execução das ferramentas de localização

A ferramenta de localização mede e identifica através do contraste de borda (que pode ser calibrado pelo usuário) a posição do identificador utilizado. Esta ferramenta é definida em uma determinada região, podendo ser um retângulo, esfera, curva ou linha.

A ferramenta de padrão geométrico faz uma varredura na imagem procurando o padrão formato pelos pixels pretos do objeto a ser rastreado. Esse padrão é calibrado diversas vezes, com iluminação uniforme e regular, antes de iniciar o seu uso. A posição do objeto é dada pelo cálculo do epicentro do padrão de pixels encontrado. Esta ferramenta pode, inclusive, dar a rotação do objeto em relação ao seu centro ou em relação a um ponto fixo (Figura 4.5), como a barra a esquerda.

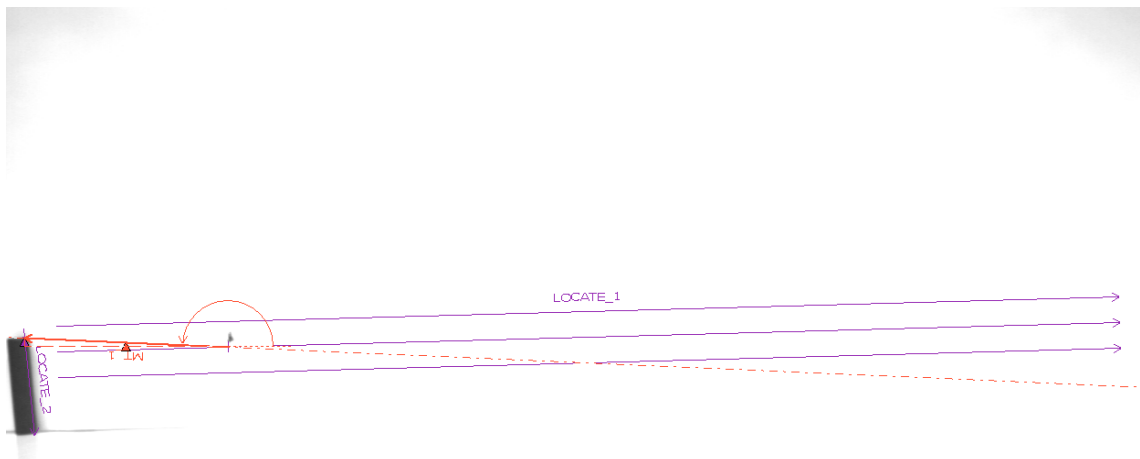
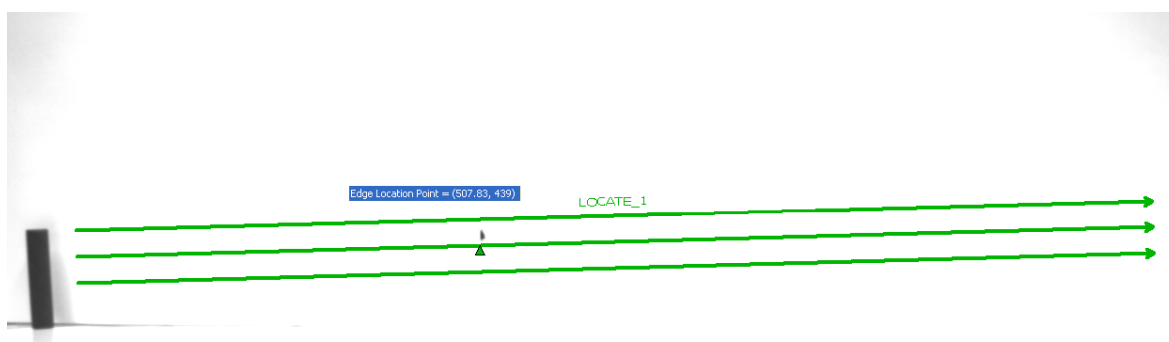


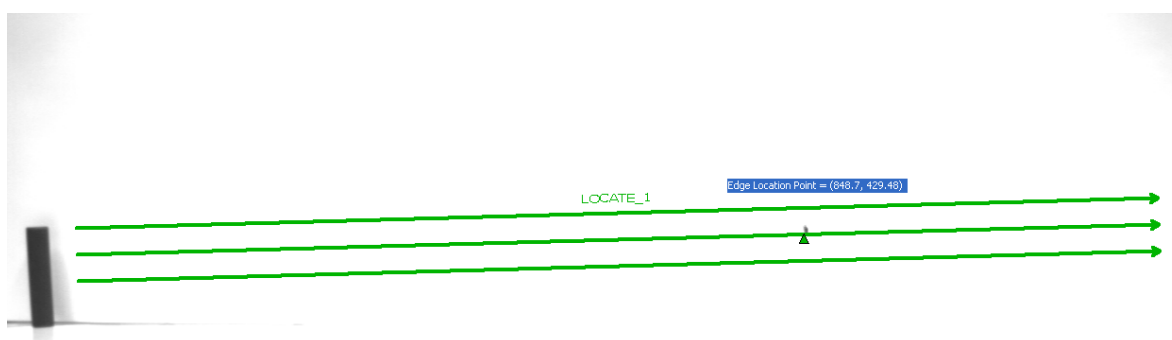
Figura 4.5 - Ferramenta de localização geométrica

Apesar da grande utilidade desta ferramenta, ela tem uma precisão ligeiramente inferior à ferramenta de localização. Deste modo, foi utilizada a ferramenta de localização como principal e a ferramenta de padrão geométrico como uma redundância, já que as ferramentas possuem uma *flag* de falha que possibilita o seu descarte e o uso da medição da outra ferramenta.

Após a escolha e implementação das ferramentas é possível fazer a sua calibração, com uma série de fotos ou com um vídeo. Dois testes da calibração (feita por fotos) são ilustrados a seguir, na Figura 4.6.



(a) Localização na posição 1



(a) Localização na posição 2

Figura 4.6 – Teste das ferramentas de localização

O software identifica a posição na matriz de pixels e informa suas coordenadas  $x$  e  $y$ . Para visualizar esses pontos e plotar um gráfico da posição da ponta esses dados foram passados pro Matlab, através de uma rede Ethernet. No Matlab é possível abrir uma porta para comunicação, a utilizada foi a 20000, e com o IP da máquina pode-se carregar os dados da câmera. Para poder tratar os dados é utilizado um identificador de início de frame (“a”) e um de final (“b”). Os frames chegam da seguinte forma (a12.01 , 33.11b). No Matlab esses identificadores são removidos de modo a termos apenas as coordenadas  $x$  e  $y$  da ponta do barbante.

Esta câmera, quando utilizada para processar os dados, faz a captura dos frames a partir de um acionamento externo, chamado de *Trigger*. Para se obter uma amostragem contínua foi utilizado o CLP, com uma rotina simples no *main program*, para tirar 8 fotos por segundo, ou seja, uma taxa de amostragem de 125 ms de modo que o Matlab consiga receber os dados sem perdas consideráveis. A conexão do CLP com a câmera foi feita através de um módulo de saída digital conectado à câmera por um cabo específico, parte integrante do kit da câmera.

#### 4.4 Análise dos resultados

Primeiramente foi feita a análise da movimentação do carro com velocidade constante (Figura 4.7), para observar a oscilação do barbante durante uma movimentação um pouco mais brusca. Para a escolha da velocidade a ser utilizada, foi feito o cálculo da velocidade média utilizando o perfil de velocidade, de modo que as duas trajetórias demorassem aproximadamente o mesmo tempo.

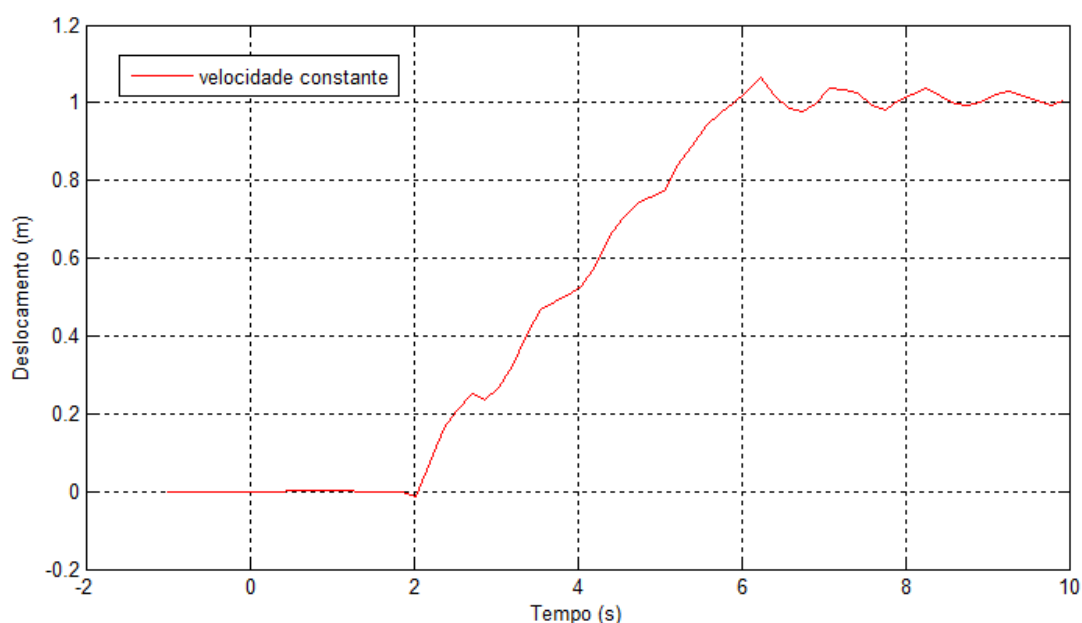


Figura 4.7 – Ponta do barbante durante a movimentação com velocidade constante

É possível observar que a ponta do barbante oscila bastante durante a trajetória e ao final da movimentação. Após alguns testes constatou-se que o barbante oscila com uma

frequência padrão e, a medida que se desloca, a amplitude desta oscilação vai diminuindo até que o carro para e ela volta a aumentar. A título de comparação a Figura 4.8 a seguir mostra a trajetória da ponta do barbante com a velocidade constante sobreposta a saída desejada, referente à simulação.

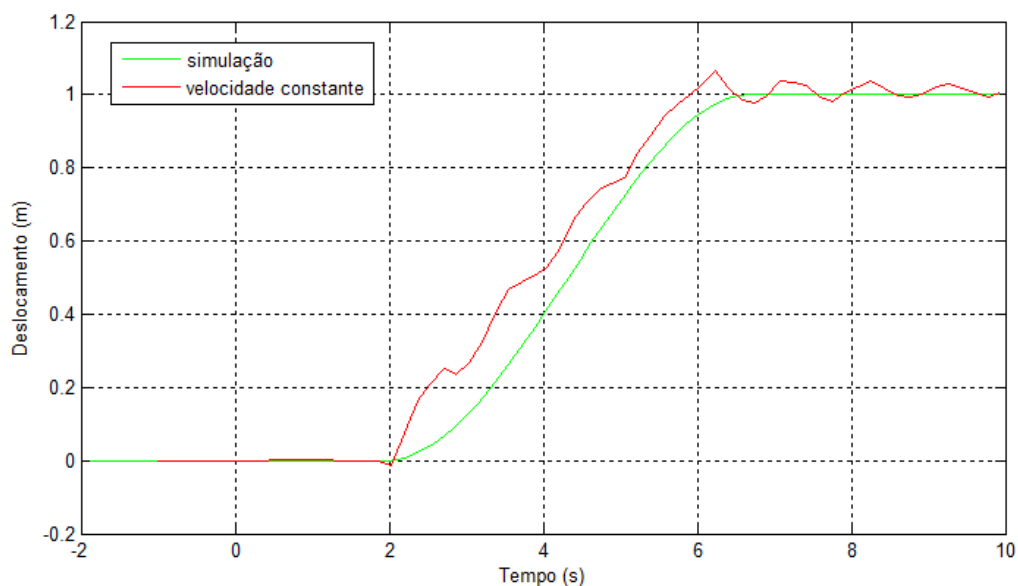


Figura 4.8 – Comparação entre saída com o carro em velocidade constante e a simulada

O erro em regime para o caso com a velocidade constante chega a ser de 6,01% ou 6,01 cm, no momento em que o carro para. Além disso, durante toda a trajetória a posição da ponta do barbante está sempre muito longe da desejada.

Posteriormente, foi feito o monitoramento com a execução do perfil de velocidade proposto (Figura 4.9).

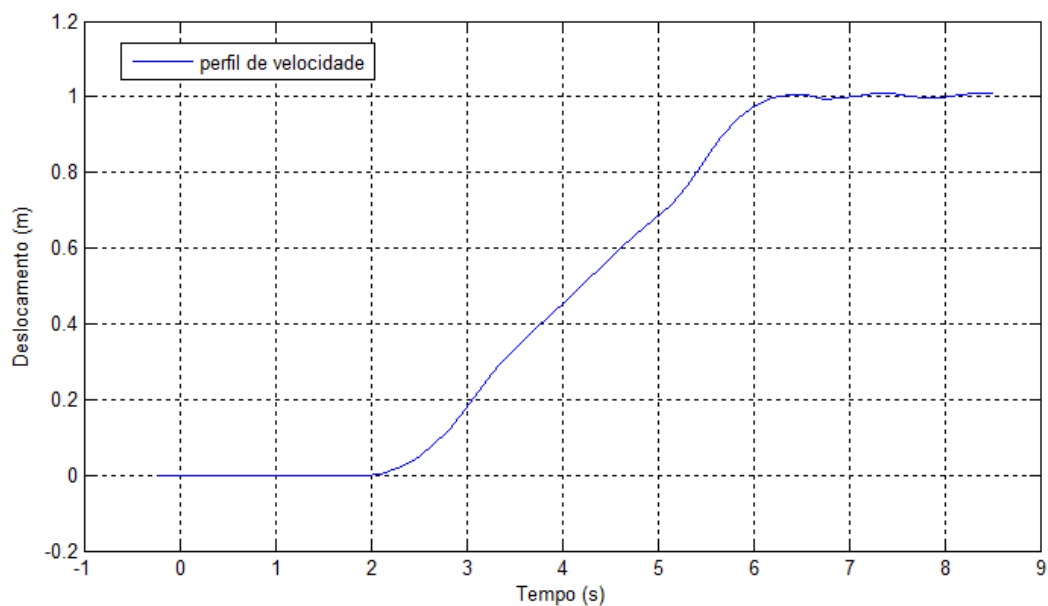


Figura 4.9 – Perfil de velocidade medido pela câmera



Para comparar a saída real (perfil de velocidade) com a desejada (simulação), foi construído o gráfico a seguir (Figura 4.10), sobrepondo a simulação realizada com a posição identificada pela câmera.

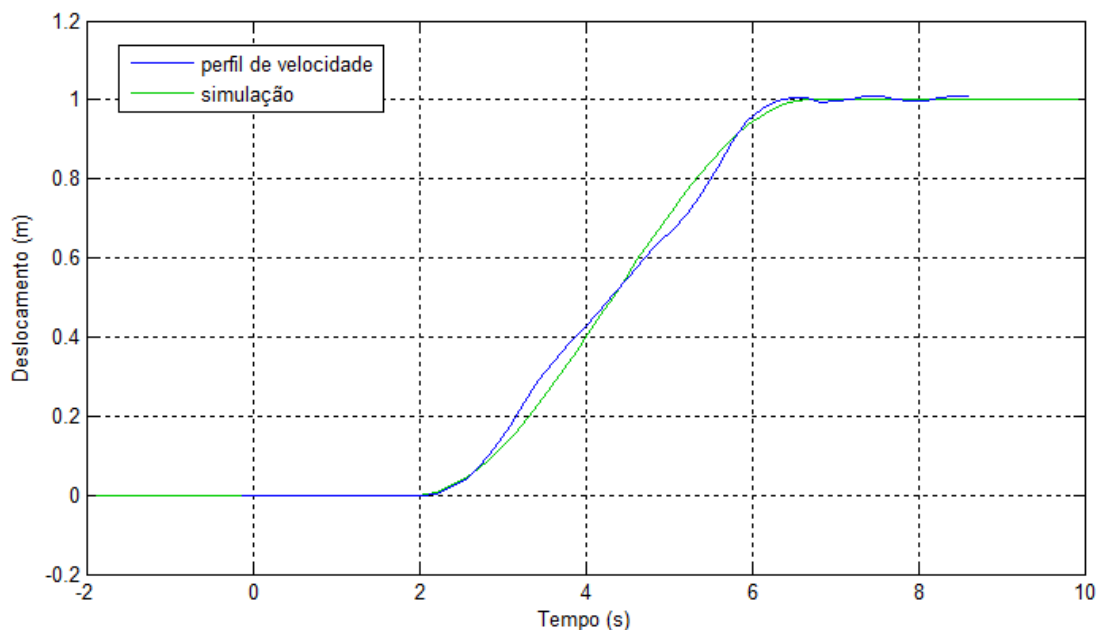


Figura 4.10 – Comparação entre saída real e saída desejada

É possível observar que a proximidade entre as duas curvas é bem grande, tanto no final, momento propício para oscilação da ponta do barbante, quanto durante a trajetória. O erro foi de 0,6 % na oscilação final, ou de 0,6 cm. Após vários testes o erro médio foi de 0,9 cm. Existem outras duas regiões com erros acentuados, sendo uma na primeira metade do movimento, em que o barbante fica a frente da posição esperada, e uma na segunda metade em que ele permanece atrás da posição simulada. O erro máximo nesses pontos foi de 4,15 cm e 5,6 cm, respectivamente.

## 4.5 Conclusão

Apesar das simplificações feitas e das peculiaridades do experimento, os resultados foram consideravelmente próximos do esperado. O erro no final da trajetória se deve, principalmente, ao não cumprimento do perfil de velocidade imposto de forma precisa, como mostrado na Figura 3.39 – do *Trend* do motor. Já o erro durante a execução da trajetória pode ter sido ocasionado devido à imprecisão do modelo para pequenos comprimentos de barbante e imprecisão no cálculo do arrasto do ar. Uma saída é calcular o arrasto de forma experimental, entretanto o tamanho reduzido da ponte rolante disponibilizada não permitiu aferir de forma empírica o arrasto no ambiente do experimento.

Apesar dos erros medidos observados durante a trajetória, do ponto de vista de acompanhamento de trajetória estes erros são pequenos. Os dados são passados para o computador em tempo real, através da rede ethernet entre a câmera e o computador. Desta forma, é possível passar através do cabo RS232 ou da rede Ethernet uma informação para o CLP como, por exemplo, um sinal de realimentação. Uma lei de controle para acompanhamento de trajetória pode ser implementada nesta bancada, para corrigir o posicionamento da ponta do barbante de forma sub-atuada, alterando a velocidade do carro. Uma rede OPC entre o Linx e o CLP já foi criada e iniciaram-se os testes para trabalhar em malha fechada.

Conclui-se que a aplicação desenvolvida, caso seja aprimorada, pode ser aplicada a indústria offshore poupando muito tempo nas operações de reentrada de riser e aumentando a confiabilidade e segurança dessas operações.

# REFERÊNCIAS BIBLIOGRÁFICAS

[1] SARPKAYA, T. Force on a circular cylinder in viscous oscillatory flow at low Keulegan-Carpenter numbers. *Journal of Fluid Mechanics*, v. 165, p. 6171, 1986.

[2] LEVINE, J. *Analysis and Control of Nonlinear Systems: A Flatness-based Approach*. [S.l.]: Springer, 2009. (Mathematical Engineering). ISBN 9783642008399.

[3] FORTALEZA, E. L. F. *Active Control Applied to Offshore Structures: Positioning and Attenuation of Vortex Induced Vibrations*. Tese (Doutorado) Ecole Nationale Supérieure des Mines de Paris, 2009.

[4] DAHLEH M.; DAHLEH, M. V. G. *Lectures on dynamic systems and control*. Chapter 12: Modal Decomposition of State-Space Models. 2011.

[5] ROWELL, D. *Modal Decomposition and the Time-Domain Response of Linear Systems*. 2004.

[6] MONTEIRO, F. R. *Controle por planejamento e rastreamento de trajetória para risers marinhos verticais*. Brasília, Brasil, Projeto de Graduação em Engenharia Mecatrônica. Universidade de Brasília, 2014.

[7] ALBUQUERQUE, D. G. *Modelagem e controle de risers em operações offshore*. Brasília, Brasil, Projeto de Graduação em Engenharia Mecânica. Universidade de Brasília. 2012.

[8] Auto Tuning Tips - Brandon Bernier - Servo Motion Control and Services Specialist.

[9] MONTEIRO, F.R., et al. *Modal Reduction Based Tracking Control for Installation of Subsea Equipment*.

[10] Rockwell Automation Publication (2012). *User Manual – Kinetix 6000 Multi-axis Servo Drives- 2094-UM001G-EN-P*.

[11] ControlLogix® 5560M03SE Combination Controller and SERCOS interface - *Installation Instructions*.

[12] Logix5000 Controllers General Instructions Reference Manual - *Reference Manual*.

[13] Logix5000 Controllers Advanced Process Control and Drives Instructions - *Reference Manual – Allen Bradley*.

- [14] Logix5000™ Controllers Motion Instructions – Allen Bradley.
- [15] Rockwell Automation Publication (2012). Instalation Instructions – Line Interface Module - 2094-IN005H-EN-P.
- [16] PresencePLUS® Pro/ProII/ProII 1.3/ProII Color – User's Manual – Banner.
- [17] <http://mc-moto.blogspot.com.br/2013/04/modelo-osi.html>, acessado em 22/11/2014.
- [18] Rockwell Automation Publication (2012). User Manual – Kinetix 6000 Multi-axis Servo Drives- 2094-UM001G-EN-P.
- [19] <http://www.petrobras.com.br/pt/nossas-atividades/areas-de-atuacao/exploracao-e-producao-de-petroleo-e-gas/pre-sal>, acessado em 21/11/2014.
- [20] SERCOS and Analog Motion Configuration and Startup – User's Manual