

MINISTERIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
ESCOLA DE ENGENHARIA

Programa de Pós-graduação em Engenharia Mecânica - PPMEC

**Desenvolvimento de um Sistema Mecatrônico para a Manutenção do
Arco Elétrico de Sistemas do Tipo MIG/MAG em Processos de
Soldagem de Juntas em Configuração T**

Leonardo Rivas Arce – 100919

Professor orientador: Dr. Douglas Bezerra de Araújo

Rio Grande – RS, Brasil

20 Fevereiro de 2015, v-1.3

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
ESCOLA DE ENGENHARIA

Programa de Pós-Graduação em Engenharia Mecânica - PPMec

**Desenvolvimento de um Sistema Mecatrônico para a Manutenção do
Arco Elétrico de Sistemas do Tipo MIG/MAG em Processos de
Soldagem de Juntas em Configuração T**

Leonardo Rivas Arce – 100919

Trabalho realizado no Laboratório de Pesquisa em Engenharia da Soldagem - LAPES da Escola de Engenharia da FURG, dentro do Programa de Pós-graduação em Engenharia Mecânica – PPMec, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Mecânica.

Orientador: Dr. Douglas Bezerra de Araújo

Rio Grande - RS, Brasil

20 de Fevereiro de 2015, v-1.3

Leonardo Rivas Arce

Desenvolvimento de um Sistema Mecatrônico para a Manutenção do Arco Elétrico de Sistemas do tipo MIG/MAG, em Processos de Soldagem de juntas em configuração T

Dissertação apresentada à Universidade Federal do Rio Grande – FURG, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Mecânica.

Área de concentração: Engenharia de Fabricação

Data: 20/02/2015

Resultado: Aprovado

Banca Examinadora:

Douglas B. de Araújo
Orientador: Professor Douglas Bezerra Araújo, Dr. (PPMec/FURG)

Carlos Eduardo Pereira
Professor Carlos Eduardo Pereira, Dr. (PPGEE/UFRGS)

Vagner Santos da Rosa
Professor Vagner Santos da Rosa, Dr. (C3/FURG)

Luciano Volcanoglo Biehl
Prof. Dr. Luciano Volcanoglo Biehl
Coordenador do PPMec/FURG

*... to my wife
Karina Valle Diaz*

Agradecimentos

Os agradecimentos principais são direcionados ao Ministério de Ciência e Tecnologia da Costa Rica – (MICIT) e ao Instituto Tecnológico da Costa Rica - ITCR e demais entes e pessoas que contribuíram neste trabalho.

Agradecimentos especiais à OEA-PAEC, CAPES pelo financiamento da bolsa de estudos.

Leonardo Rivas Arce

Desenvolvimento de um Sistema Mecatrônico para a Manutenção do Arco Elétrico de Sistemas do Tipo MIG/MAG em Processos de Soldagem de Juntas em Configuração T/ Leonardo Rivas Arce. – Rio Grande - RS, Brasil, 20 de fevereiro de 2015, v-1.3.

246 p.: il. (algumas color.); 30 cm.

Orientador: Dr. Douglas Bezerra de Araújo

Dissertação (Mestrado) – MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
ESCOLA DE ENGENHARIA

Programa de Pós-Graduação em Engenharia Mecânica - PPMec, 20 de fevereiro de 2015, v-1.3.

1. Soldagem. 2. Mecatrônica. I. Dr. Douglas Bezerra de Araújo.
II. Universidade Federal de Rio Grande - FURG. III. Escola de Engenharia.

CDU 00:000:000.0

Resumo

A importância da automação da manutenção do arco elétrico nos processos de soldagem possui várias arestas que são a simplificação, a erradicação dos erros inseridos pelos operários no posicionamento da tocha, o aumento da produtividade ao acelerar o processo de produção, aumento na qualidade ao manter a posição relativamente estável e assim interferir minimamente nas variáveis importantes do arco elétrico na soldagem que são o comprimento do arco, velocidade da soldagem, que são variáveis altamente ligadas à eficiência na transferência de potência da fonte de soldagem até as peças a serem soldadas.

O desenvolvimento de um sistema mecatrônico para conseguir manter o arco elétrico ao longo do processo de soldagem MIG/MAG é um processo complexo que precisa o envolvimento interdisciplinar de várias áreas do conhecimento, como é a computação, a eletrônica, a mecânica, entre muitas outras. Esse desenvolvimento é apresentado neste trabalho, cuja abordagem abrange metodologias de implantação e projeto que permitirão a integração unificada da eletrônica e da mecânica para obter um sistema mecatrônico.

A análise desde o ponto de vista mecatrônico foi priorizado dado que o sistema não é um simples aparelho mecânico ou um simples sistema eletrônico, aliás é um produto que integrou diversas áreas do conhecimento, além de múltiplas ferramentas de software na etapa de projeto, tais como Matlab, Solidworks, Cadsoft Eagle, PICC, MPLAB, NI Multisim, P-Spice, Octave, Microbrn, e diversas máquinas de fabricação mecânica como são o torno, fresa, retíficas e furadeiras, cujo projeto devia se basear na possibilidade de utilizar estes equipamentos e programas.

Uma particularidade do sistema desenvolvido foi o mecanismo e automação do procedimento de posicionamento da tocha de soldagem e a manutenção desta posição ao longo do processo, além do controle da velocidade da soldagem na direção da geração do cordão de solda, estabilizando o arco elétrico e mantendo-o acendido ao longo do processo de soldagem. Essa automação foi possível graças à utilização de sensores de distância, por ultrassom e infravermelhos, além do sensoramento da tensão e corrente dos diferentes atuadores elétricos do sistema. Foi necessário a criação e implantação de algoritmos computacionais e de controle para conseguir a automação deste processo, além do projeto e fabricação das peças mecânicas necessárias baseado na metodologia de desenvolvimento CAD, priorizando sempre a funcionalidade do mecanismo a se implantar.

Palavras-chaves: Soldagem; Mecatrônica; Automação.

Abstract

The automation process of keeping arc welding on, throughout a welding process contain many edges, including process simplification, error removal placed by users at torch positioning, increasing of productivity and quality maintain torch position relatively stable, with a minimal disturbing in the most important arc welding variables. These variables are arc length, welding speed, extremely related to power efficiency delivery from welding power source to workpieces.

The mechatronic device development to keep arc welding throughout a MIG/MAG welding process it is a complex process, implying an interdisciplinary work, in which areas such as computing, electronics and mechanics was involved. The designing, implementation and validation process of this mechatronic device is showing in this work, in with the approach, implementation and design methods used allows a unified integration of electronics and mechanics, in other works mechatronics.

Analytics mechatronics point of view was fundamental, because the project implemented is not about a simple mechanic device or a simple electronic system, moreover is a product integrating multidisciplinary fields. In addition numerous software utilities such as, Matlab, Solidworks, Cadsoft Eagle, PICC, MPLAB, NI Multisim, Octave, Microbrn, C++ was used, and several mechanicals machines including lathe, milling, grinders and drills, whose design has to complaint the possibility to use this equipment's and software.

A system development particularity was the positioning automation proceed of welding torch, and maintain it stable throughout the welding process, also keeping constant and stable the welding speed in the welding cord generation direction. The automation of torch positioning was possible due to ultrasonic and infrared distance sensors used for this purpose, measuring as well voltage and current of electrical system actuators. Was necessary the generation and implementation of control and computational algorithms to reach the automation of arc welding stabilization, as well as designing and manufacturing mechanical parts based on CAD development, prioritizing always the mechanism functionality.

Keywords: Welding; Mechatronics; Automation.

Resumen

La importancia del control automatizado en el mantenimiento del arco eléctrico en el proceso de soldadura, posee diferentes características que son la simplificación del proceso, eliminación de los errores humanos al posicionar la antorcha, el aumento de la productividad al acelerar la producción, incremento en la calidad al mantener la posición relativamente estable, y de esta forma reducir al mínimo la interferencia en las variables importantes del arco eléctrico en el proceso de soldadura, que son longitud del arco eléctrico, la velocidad del proceso de soldadura, que son variables en extremo importantes, asociadas a la eficiencia energética en la transferencia de potencia entre la fuente del equipo de soldadura y las piezas que serán soldadas.

El proceso de desarrollo de un sistema mecatrónico para lograr mantener el arco eléctrico a través de un proceso de soldadura del tipo MIG, es un proceso de alta complejidad que demanda la interacción multidisciplinaria de diversas áreas del conocimiento, como lo son la ingeniería de software, electrónica y mecánica, etc. Ese desarrollo es mostrado en el presente trabajo, cuyo abordaje involucra metodologías de implementación y diseño que permitirán la integración unificada de la electrónica y mecánica para crear un sistema mecatrónico.

El análisis realizado fue hecho desde el punto de vista mecatrónico, sabiendo que el sistema no es tan solo un mecanismo o un sistema electrónico, por el contrario es un producto creado matemáticamente integrando diversas áreas del conocimiento, además de múltiples programas en la etapa de diseño, tales como Matlab, Solidworks, Cadsoft Eagle, PICC, MPLAB, NI Multisim, P-Spice, Octave, Microbrn, e diferentes máquinas de fabricación mecánica como lo son el torno, la fresadora, rectificadora y taladros, onde el diseño a realizar debía basarse en la posibilidad de usar estos dispositivos y herramientas de software.

Una singularidad del sistema desarrollado fue la automatización del procedimiento de posicionamiento de la antorcha de soldadura, y la corrección de esa posición a lo largo del proceso, además del control automático de la velocidad del proceso en la ruta de generación del cordón de soldadura. Esta automatización fue posible debido al uso de sensores de distancia por infrarrojo y por ultrasonido, además de la medición de la corriente y tensión eléctrica de los diferentes actuadores del sistema. Es claro que fue necesario la creación e implementación de diferentes algoritmos computacionales y de control, para lograr la automatización del proceso, por otra parte también para la fabricación de los componentes mecánicos necesarios usando la metodología de diseño asistido por computadora CAD, dando prioridad a la funcionalidad del mecanismo a construir.

Palabras Clave: Proceso de Soldadura; Mecatrónica; Automatización.

Lista de Tabelas

Tabela 3.2.1 Componentes e equipamentos do Laboratório de Pesquisa em Engenharia de Soldagem (LAPES).	31
Tabela 3.2.2. Componentes utilizados para o projeto.....	32
Tabela 4.1.1. Dimensões dos componentes principais do sistema mecatrônico.	35
Tabela 4.2.1. Velocidades utilizadas para o projeto mecânico.....	38
Tabela 4.2.1. Constantes definidas para o cálculo dos esforços da Fig. 4.2.3.....	40
Tabela 4.2.2. Parâmetros do motor de corrente contínua.	43
Tabela 4.5.1. Componentes e peças alojadas no mecanismo angular.	53
Tabela 5.1.1. Principais características do sensor US-100.	58
Tabela 5.2.1. Valores utilizados do Ciclo de Trabalho para os servos-motores.....	67
Tabela 5.2.2. Medições de velocidade do motor para o cálculo da constante K_e	70
Tabela 5.2.3. Atrasos nas medições de velocidade no encoder.	76
Tabela 5.2.4. Parâmetros de projeto do sistema de controle de velocidade.	79
Tabela 5.2.5. Constantes de compensação para controle de perturbações do motor em tempo contínuo.	81
Tabela 5.2.6. Resumo da resposta ao salto do sistema compensado da figura 5.2.20.....	82
Tabela 5.2.7. Parâmetros do MOSFET 2N60.....	91
Tabela 5.2.8. Parâmetros dos transistores BJT.....	93
Tabela 5.2.8. Constantes medidas no robô para o controle do ângulo de soldagem no posicionamento da tocha.	103
Tabela 5.2.9. Constantes medidas no robô e parâmetros gerais do sistema angular.	104
Tabela 5.2.10. Valores obtidos no sensor GP2Y0A21YK após filtragem.	111
Tabela 5.2.11. Características do driver L6203.....	112
Tabela 5.2.12. Sequência de passos implantada no microcontrolador PIC18F4550 para evitar curto-circuito na saída do L6203.....	118
Tabela 5.2.13. Parâmetros definidos para a temporização na intercomunicação dos subsistemas.	121
Tabela 5.2.14. Comandos definidos para a intercomunicação dos subsistemas.....	121
Tabela 5.2.15. Descrição dos sinais de entrada do sistema para o correto funcionamento da UPC.....	123
Tabela 6.5.1. Resumo dos pulsos medidos no sensor de velocidade vistos na Fig. 6.5.1	134
Tabela 6.5.2. Resumo dos pulsos medidos no sensor de velocidade vistos na Fig. 6.5.2.	135
Tabela 6.5.3. Valores limites de assentamento do sistema com o tempo de ajuste do ciclo de trabalho.	135
Tabela 6.6.1. Resumo do algoritmo de cálculo da equação de diferenças dentro do microcontrolador para compensação nas variações da corrente, parte 1.....	142
Tabela 6.6.2. Continuação da Tab. (6.6.1).	143

Lista de Figuras

Figura 1.1.1. Sistema robotizado de soldagem de navios.....	3
Figura 1.2.1. Graus de liberdade do sistema mecânico a projetar.	4
Figura 2.1.1. Robô guiado por sensoriamento (varredura da face).	7
Figura 2.1.2. Programação automatizada (testes na face) numa peça onde a superfície deve ser refeita com soldagem.	8
Figura 2.1.3. Sistema mecatrônico para soldagem de pinos metálicos.	9
Figura 2.1.4. Unidade mecânica proposta para o sistema de monitoramento de multipasses de soldagem.	10
Figura 2.1.5. Esquemático do sistema de visão com o anteparo.	10
Figura 2.1.6. Sistema Tartílope V2.0 utilizado para o projeto do Sistema de Seguimento de Junta (SSJ).	12
Figura 2.1.7. Sistema Tartílope V4.0.	13
Figura 2.1.8. Sistema robótico com seis graus de liberdade para soldagem de navios com paredes duplamente reforçadas.....	14
Figura 2.1.9. Sistema robótico inteligente para soldagem de navios (a) Robô de soldagem (b) Esquemático do processo.....	15
Figura 2.2.1. Circuito fechado de tensão e corrente na soldagem.	18
Figura 2.2.2. Curvas características de tensão e corrente na soldagem para diferentes DBCP.....	19
Figura 2.2.3. Potencial elétrico das regiões que conformam o arco elétrico de soldagem (a) cátodo (b) Coluna de plasma (c) ânodo.	19
Figura 2.2.4. Representação simplificada das direções dominantes de fluxo de calor nas diferentes regiões do arco.	21
Figura 2.2.5. Comportamento da corrente e da tensão médias do arco elétrico para alterações da DBCP num ensaio robotizado.....	22
Figura 2.2.6. Rendimento de fusão (a) baixa relação Calor/velocidade de soldagem (b) Alta relação Calor/Velocidade (c) Curva de rendimento-parâmetro adimensional $\eta_{UIV}/(Hav)$	22
Figura 2.2.7. Eficiência do arco elétrico dos processos SAW e GMAW(MIG/MAG).....	23
Figura 2.3.1. Ocorrência das distorções numa barra de aço (a) aquecendo e resfriando sem restrições físicas (b) restringindo a expansão durante o aquecimento produz uma deformação da barra ao se expandir onde não for restringida. (c) deformação ocorrida na barra ao resfriar.....	24
Figura 2.3.2. Exemplo da distorção numa soldagem de chapas em junta T.....	25
Figura 2.4.1. Aplicação do sensor ultrassônico no processo de soldagem MIG/MAG.....	26
Figura 2.4.2. Medição da intensidade luminosa do arco elétrico na soldagem com filtragem e sem filtragem.....	27
Figura 2.5.1. Fonte da soldagem MIG com transformador com derivação central.	28
Figura 2.5.2. Fonte da soldagem de última geração com sensoriamento e controle baseado nas variáveis de tensão e corrente do arco.....	28
Figura 3.1.1. Espectro de magnitude de Fourier para o conjunto tensão-corrente da soldagem MIG/MAG medida e manipulada com o robô da MOTOMAN.	30
Figura 3.2.1. Ferramentas do LAPES e algumas adquiridas para o projeto.....	33
Figura 4.1.1. Interface do usuário para projeto o mecânico das dimensões dos sistemas mecânicos e o cálculo dos deslocamentos.....	36

Figura 4.2.1. Par cinemático plano-junta dos elos do sistema de velocidade constante, (a) Esquemático (b) Diagrama equivalente.....	38
Figura 4.2.2. Desenho do sistema mecânico de velocidade constante com o Solidworks®, as dimensões estão em milímetros.	39
Figura 4.2.3. Diagrama de forças para o cálculo do motor e a relação das polias-correia.	40
Figura 4.2.4. Polia escolhida para montagem no eixo das rodas. (a) Dimensões da polia em mm (b) apresentação do espaço disponível para a montagem da polia.	42
Figura 4.4.1. Diagrama movimento de 4 elos e 4 barras com um grau de liberdade.	44
Figura 4.4.2. Diagrama de corpo livre para o cálculo dos esforços e do motor.	45
Figura 4.4.3. Desenho em 3D do mecanismo da direção com Solidworks®.....	46
Figura 4.4.4. Diagrama de corpo livre do sistema da direção em 3D.	47
Figura 4.4.5. Diagrama de corpo livre do eixo vertical do mecanismo da direção.	48
Figura 4.5.1. Mecanismo angular desenhado no Solidworks®.....	52
Figura 4.5.2. Diagrama de corpo livre do mecanismo angular e dos centros instantâneos.	52
Figura 4.5.3. Parâmetros do parafuso sem fim para transmissão da potência.	54
Figura 4.5.4. Sistema mecânico completo em 3D, montado no Solidworks®.....	55
Figura 5.1.1. Esquema da distorção das peças metálicas ao longo do processo de soldagem que se tentará corrigir.	57
Figura 5.1.2. Posicionamento dos sensores de distância por ultrassom para correção da trajetória do robô (vista superior do sistema).	59
Figura 5.1.3. Diagrama para o cálculo da distância mínima de posicionamento dos sensores US-100.	59
Figura 5.1.4. Esquema proposto para correção da trajetória e posição, (a) não há distorção ou deformação para corrigir, (b) Correção positiva (c) Correção negativa.	60
Figura 5.1.5. Variância máxima da posição do bico de contato ao detectar distorção ou deformação da trajetória linear.	61
Figura 5.2.1. Diagrama de blocos do sistema proposto.....	62
Figura 5.2.2. Diagrama de temporização do sensor de distância US-100, usado para medir a distância do robô à peça.....	63
Figura 5.2.3. Algoritmo computacional para o cálculo da correção na trajetória e posição baseado em sensoriamento por ultrassom, programado no PIC16F877A.....	64
Figura 5.2.4. Diagrama de blocos do subsistema atuador.	65
Figura 5.2.5. Esquemático da configuração do oscilador interno do PIC10F322.....	67
Figura 5.2.6. Circuito eletrônico para o sistema atuador de controle dos servos.	68
Figura 5.2.7. Layout do circuito impresso para o sistema atuador de controle dos servos.	68
Figura 5.2.8. Algoritmo computacional programado no PIC10F322.....	69
Figura 5.2.9. Corrente de rompimento da zona morta no motor de corrente contínua.....	71
Figura 5.2.10. Medição da indutância do motor com o LCR.	71
Figura 5.2.11. Medição da constante de tempo mecânica do motor com o osciloscópio.....	72
Figura 5.2.12. Gráfico para obtenção do coeficiente de viscosidade mecânica.	74
Figura 5.2.13. Resposta ao salto para o motor de CC em laço aberto.....	74
Figura 5.2.14. Resposta ao impulso para o motor de CC em laço aberto.....	75
Figura 5.2.15. Resposta ao salto em laço aberto para o motor de CC com o atraso do sensor para a velocidade menor.....	77
Figura 5.2.16. Resposta ao impulso em laço aberto para o motor de CC com o atraso do sensor para a velocidade menor.....	77
Figura 5.2.17. Resposta ao salto em laço aberto para o motor de CC com o atraso do sensor para a velocidade maior.....	78
Figura 5.2.18. Resposta ao impulso para o motor de CC com o atraso do sensor para a velocidade maior.....	78

Figura 5.2.19. Resposta ao salto para o motor de CC em laço fechado com ganância unitária.....	80
Figura 5.2.20. Resposta ao salto para o sistema em laço fechado e compensador contínuo....	81
Figura 5.2.21. Diagrama de blocos do motor com feedback.....	83
Figura 5.2.22. Resposta ao salto do sistema realimentado em tempo discreto, com um tempo de amostragem de 300 μ s.....	84
Figura 5.2.23. Diagrama de blocos do motor com o feedback do sensor no domínio da frequência.....	85
Figura 5.2.24. Resposta ao salto em laço fechado para o motor utilizando o sensor com o atraso na velocidade menor.....	85
Figura 5.2.25. Forma de onda do sinal PWM.....	87
Figura 5.2.26. Resposta em frequência do motor de corrente contínua.....	89
Figura 5.2.27. Sinal PWM teórica percebida pelo motor com uma frequência de 30 kHz e um ciclo de trabalho do 50%.....	90
Figura 5.2.28. Diagrama geral do sistema gerador de PWM para o motor de CC.....	92
Figura 5.2.29. Circuito driver do MOSFET.....	92
Figura 5.2.30. Rede SNUBBER para proteção do MOSFET.....	94
Figura 5.2.30. Sistema Gerador do PWM completo.....	95
Figura 5.2.31. Simulação do sistema gerador PWM com um ciclo de trabalho de 68%, (a) Sinal no osciloscópio que entra no motor, PWM (vermelho), <i>Im</i> (azul); (b) Tensão média do motor.....	96
Figura 5.2.32. Simulação do sistema gerador PWM com um ciclo de trabalho do 1%, (a) Sinal no osciloscópio que entra no motor, PWM (vermelho), <i>Im</i> (azul); (b) Tensão média do motor.....	97
Figura 5.2.33. Circuito regulador PWM com MOSFET 2N60 montado no Cadsoft Eagle V5.0.....	98
Figura 5.2.34. Layout do circuito impresso para o regulador PWM com o MOSFET 2N60 montado no Cadsoft Eagle V5.0.....	98
Figura 5.2.35. Modelagem geométrica do sistema angular para a obtenção das equações de controle numérico, (a) Posição inicial (b) Posição final.....	99
Figura 5.2.36. Geometria do robô para obtenção do ângulo de soldagem baseado na distância do mecanismo à peça vertical.....	101
Figura 5.2.37. Função seno e cosseno do sistema angular para o cálculo das aproximações dos deslocamentos.....	105
Figura 5.2.38. Função ângulo de soldagem para o início do processo de soldagem.....	105
Figura 5.2.39. Função deslocamento linear para atingir o ângulo de soldagem.....	106
Figura 5.2.40. Função deslocamento linear para atingir a junta das peças metálicas.....	106
Figura 5.2.41. Diagrama geral do controle para os motores de passo no sistema angular.....	107
Figura 5.2.42. Características de saída do sensor GP2Y0A21YK da SHARP.....	108
Figura 5.2.43. Tensão de saída do sensor GP2Y0A21YK da SHARP medida no osciloscópio.....	109
Figura 5.2.44. Código de programação em linguagem C correspondente ao bloco Filtro PB para o processamento do sinal fornecida pelo sensor GP2Y0A21YK da SHARP.....	109
Figura 5.2.45. Calibração e sintonização do filtro passa baixa digital para o sensor GP2Y0A21YK da SHARP programado no microcontrolador, as medições foram feitas cada 10 mm.....	110
Figura 5.2.46. Gráfica característica do sensor GP2Y0A21YK da SHARP programado no microcontrolador.....	111
Figura 5.2.47. Sistema de aplicação proposto para o circuito integrado driver L6203.....	112
Figura 5.2.48. Motor de passo MINEBEA vendido no mercado brasileiro.....	114

Figura 5.2.49. Esquemático do circuito driver dual para o controle do motor de passo.	115
Figura 5.2.50. Layout do circuito impresso do driver dual para o controle do motor de passo.	115
Figura 5.2.51. Circuito impresso do driver dual montado para o controle do motor de passo.	116
Figura 5.2.52. Sequências dos passos para o controle dos motores MINEBEA.	116
Figura 5.2.53. Sequências dos passos para o controle dos motores MINEBEA.	117
Figura 5.2.54. Sequências dos passos para o controle implantados no microcontrolador PIC18F4550.	118
Figura 5.2.55. Algoritmo computacional programado no PIC18F4550 para o posicionamento da tocha de soldagem baseado na distância da tocha à peça.	119
Figura 5.2.56. Esquema de comunicação proposto.	120
Figura 5.2.57. Diagrama de temporização e protocolo de comunicação proposto.	121
Figura 5.2.58. Diagrama de blocos geral da UPC.	122
Figura 5.2.59. Interface de usuário definida para o sistema com os sinais necessárias para o correto funcionamento da UPC.	122
Figura 5.2.60. Diagrama de aplicação típica do MAX7219 fornecido pelo fabricante.	124
Figura 5.2.61. Circuito da interface de entrada com o usuário para o ingresso da velocidade e programação do sistema.	124
Figura 5.2.62. Layout do Circuito da interface de entrada com o usuário para o ingresso da velocidade e programação do sistema.	125
Figura 5.2.63. Circuito para a interface de saída com o usuário.	125
Figura 5.2.65. Diagrama de estados da UPC programado no PIC16F877A.	127
Figura 6.1.1. Peça da direção.	128
Figura 6.1.2. Mecanismo da direção completo.	129
Figura 6.2.1. Estrutura base do carrinho.	130
Figura 6.2.2. Mecanismos de inserção de informação para controle.	131
Figura 6.2.3. Placas de controle do sistema completo.	131
Figura 6.3.1. Mecanismos completo.	132
Figura 6.4.1. Sinal medida na entrada dos servos gerada pelo PIC10F322 para o controle da direção do robô, (a) Frequência do PWM (b) Ciclo de Trabalho do 9,1%.	133
Figura 6.5.1. Pulsos medidos do sensor de velocidade para um ciclo de trabalho do 17,6%.	134
Figura 6.5.2. Pulsos medidos do sensor de velocidade para um ciclo de trabalho do 27%. ...	136
Figura 6.6.1. Diagrama de medição da corrente do motor.	138
Figura 6.6.2. Corrente medida no motor através do resistor <i>R_{sense}</i> , (a) Torque mínimo (b) Torque máximo, em ambas figuras o PWM (amarelo), corrente (Verde).	139
Figura 6.6.3. Processo de compensação da corrente ante uma variação na carga.	143
Figura 6.7.1. Deslocamento linear do sistema angular após 7600 passos do motor, o valor medido foi de aproximadamente 38 mm.	144
Figura 6.7.2. Ângulo inicial do sistema angular antes os 7600 passos, o valor medido foi de 21°.	145
Figura 6.7.3. Ângulo final do sistema angular após os 7600 passos, o valor medido foi de 57°.	145
Figura 6.8.1. Ângulo inicial do sistema angular antes de iniciar o deslocamento após a medição da distância, o valor medido foi de 27 graus.	146
Figura 6.8.2. Distância percorrida pelo sistema para atingir o ângulo final (ou o ângulo inicial da soldagem), o valor medido foi de entre 14 e 15 mm.	147
Figura 6.8.3. Ângulo final (ou o ângulo inicial da soldagem) atingido após o deslocamento de entre 14 e 15 mm, o valor medido foi de entre 45° e 46°.	147

Figura 6.9.1. Deslocamento linear do mecanismo que suporta a tocha experimental considerando um stickout de 12mm	148
Figura 6.9.2. Distância medida do carrinho à peça (a) Medição com o paquímetro e no sensor de ultrassom (b) Dimensões do sensor IR para correção	149
Figura 6.9.3. Deslocamento linear do mecanismo que suporta a tocha.....	149
Figura 6.10.1 Envelope operacional para soldagem MAG robotizada (85% Ar + 15% CO ₂), arame de aço ao carbono 1,2 mm com peças em configuração T	150
Figura 6.10.2. Teste do sistema no processo de soldagem com um stickout de 12 mm.	150

Lista de Abreviaturas e Siglas

ABNT	Associação Brasileira de Normas Técnicas
ADC	Conversor Analógico-Digital
BMP	Imagem de Mapa de Bits
BJT	Transistor de Junta Bipolar
CAD	Computer Assisted Design
CC	Corrente Contínua
C++	Linguagem de programação computacional
CEF	Característica Estática da Fonte.
CONEM	Congresso Nacional de Engenharia Mecânica
CNC	Comando Numérico Computadorizado
DBCP	Distância do Bico de contato à Peça
CT	Ciclo de Trabalho
DTS	Datasheet
GMAW	Gas Metal Arc Welding
IEM	Interferência Eletromagnética
LAPES	Laboratório de Pesquisas em Engenharia de Soldagem
LCR	Indutímetro-Capacímetro-Ohmímetro
MAG	Metal Active Gas
MIG	Metal Inert Gas
MOSFET	Transistor de Efeito de Campo de Metal Oxido Semicondutor
PI	Proporcional-Integrativo
PWM	Modulação da largura do pulso
RDC	Resistor-Diodo-Capacitor
SSJ	Sistema Seguidor de Junta
TTL	Transistor-Transistor Lógica
UART	Unidade Assíncrona Receptor-Transmissor
UPC	Unidade de Processamento Central
US-100	Sensor de distância ultrassônico
WLRA	Welding LAPES Robotics Applications
ZAC	Zona Afetada pelo Calor
μ C	Microcontrolador

Lista de Símbolos

α	Ângulo inicial do mecanismo de posição.
β	Ângulo inicial da soldagem.
γ	Ângulo para a correção da direção do robô.
θ	Ângulo de variação máximo permitido medido na distorção.
φ	Ângulo de variação máximo do bico de contato na distorção.
h_1	Altura da mesa que suporta a tocha para um ângulo β dado.
A	Altura linear da tocha desde o centro instantâneo do mecanismo.
H	Altura do solo até o centro instantâneo do mecanismo de posição.
A_0	Área de contato entre a coluna de plasma e a peça.
t_{delay}	Atraso do sistema de medição de velocidade na direção do cordão.
C_n	Capacitância do enésimo capacitor.
$C_{SNUBBER}$	Capacitor de proteção para o driver L6203.
L	Ciclos para fadiga do rolamento.
S_e	Coefficiente de resistência mecânica a fadiga.
K_{fS}	Coefficiente de concentração de tensão pelo efeito de torção.
K_f	Coefficiente de concentração de tensão pelo efeito de flexão.
$\frac{1}{n_{seg}}$	Coefficiente de segurança pelo efeito de fadiga.
a_n	Coefficiente da série de Fourier trigonométrica.
b_n	Coefficiente da série de Fourier trigonométrica.
CT	Comprimento do eletrodo da tocha (haste metálica).
n_a	Constante arbitraria do material o tipo de gás.
k	Constante de Boltzmann $8,62 \times 10^{-5} eV/K$.
K_{tS}	Constante de concentração de tensão pelo efeito de torsão.
K_t	Constante de concentração de tensão pelo efeito de flexão.
k_a	Constante modificador da resistência à fadiga pelo material.
k_b	Constante modificador da resistência à fadiga pela geometria.
k_c	Constante modificador da resistência à fadiga pela temperatura.
k_d	Constante modificador da resistência à fadiga pelo ambiente.
k_e	Constante modificador da resistência à fadiga para confiabilidade.
k_f	Constante modificador da resistência à fadiga por fatores vários.

K_p	Constante proporcional do compensador PI.
K_I	Constante Integrativa do compensador PI.
K_e	Constante contra-electromotriz do motor de corrente continua.
t_e	Constante de tempo elétrica do motor de CC.
K_t	Constante do torque do motor de CC.
t_m	Constante de tempo mecânica do motor de CC.
C_0	Constante do patamar de carregamento do rolamento.
a,b,c	Constantes mecânicas do sistema angular.
I	Corrente do arco elétrico.
I_a	Corrente de armadura do motor de CC.
I_C	Corrente de Coletor do BJT.
I_D	Corrente do diodo (dois eletrodos) ou do Drenador do MOSFET.
I_m	Corrente de carga máxima do motor de CC.
I_S	Corrente de saturação para abrir o arco elétrico.
J	Densidade da corrente do arco.
δ	Deslocamento total da mesa linear que suporta a tocha.
$\Delta x'$	Deslocamento total do sistema mecânico angular para atingir o ângulo inicial da soldagem.
Δx	Deslocamento total do bico de contato ao longo do eixo x.
Δy	Deslocamento total do bico de contato ao longo do eixo y.
ΔDIR	Diferencial da distância medida dos sensores ultrassônicos.
b_1	Distância do deslocamento da mesa no eixo x para ângulo β dado.
D_m	Distância medida pelo sensor infravermelho.
$D_m(min)$	Distância mínima de colocação do carrinho desde a peça.
D_1	Distancia medida pelo sensor ultrassônico traseiro.
D_2	Distancia medida pelo sensor ultrassônico dianteiro.
I_{RD}	Distância do centro do bico de contato com a peça a se soldar.
η	Eficiência da fonte de potência.
Q_n	Enésimo Transistor BJT.
$m(k)$	Equação de diferenças para o controle da velocidade do motor.
$e(k)$	Erro da corrente do motor medida no μC .
$e(k\%)$	Erro medido da corrente do motor em porcentagem.
<i>erro</i>	Erro proporcional na medição de velocidade do robô.
E_b	Força contra-electromotriz do motor de corrente continua.
$\overline{F_{sal}}$	Força de saída do mecanismo da direção.

F_{ent}	Força de entrada aplicada pelo motor.
F_d	Força dinâmica do sistema.
F_s	Força estática do sistema.
F_N	Força normal.
F_e	Força radial equivalente para o rolamento.
F_r	Força radial para o rolamento.
f_{SW}	Frequência de chaveamento.
f_{PWM}	Frequência do sinal PWM.
f_{OSC}	Frequência de oscilação do microcontrolador.
$G(z)$	Função de transferência do motor em tempo discreto.
$E(z)$	Função erro do sistema em tempo discreto.
$H(s)$	Função de transferência no domínio da frequência.
$G(s)$	Função de transferência em laço fechado do motor de CC.
$C_{PI}(s)$	Função no domínio da frequência do compensador PI.
$G_{PI}(s)$	Função do compensador PI em laço fechado.
$C_{PI}(z)$	Função do compensador PI no domínio do tempo discreto.
$f(t)$	Função sinal PWM para controle da velocidade.
K_s	Ganância do sensor de velocidade.
L_a	Indutância da armadura do motor de CC.
L_n	Indutância do enésimo indutor.
q	Magnitude de carga do elétron $8,62 \times 10^{-5} eV/K$.
m	Massa específica do corpo.
J_m	Momento de Inércia do motor.
M	Momento de inércia.
M_s	Momento de inércia estático.
M_{ms}	Momento de inércia estático do eixo do motor.
M_d	Momento de inércia dinâmico.
M_{md}	Momento de inércia dinâmico no eixo do motor.
M_{servo}	Momento de Inércia no eixo do servo.
T_M	Par necessário do motor.
T_p	Período do sinal PWM.
μ_0	Permeabilidade magnética.
x_c	Pontos no eixo x na trajetória circular desde o centro instantâneo.
y_c	Pontos no eixo y na trajetória circular desde o centro instantâneo.

x_2	Ponto final no eixo x para o posicionamento da tocha de soldagem.
V_T	Potencial térmico do material dielétrico o do gás.
P_D	Potência do Drenador do MOSFET.
P_{Rn}	Potência do enésimo resistor.
P_a	Pressão atmosférica.
P	Pressão do jato de plasma.
P_r	Pressão radial para o rolamento.
V_m	Queda de tensão do motor de CC (corrente contínua).
V_D	Queda de tensão do diodo de rolamento livre.
R_{plasma}	Raio maior da coluna de plasma.
r_{ar}	Raio do arame.
$r_{crit.}$	Radio crítico do eixo cilíndrico para fadiga.
r_{roda}	Raio da roda do carrinho.
r	Raio do bico de contato da tocha.
R	Raio maior da tocha de soldagem.
R_a	Resistência da armadura do motor de CC.
R_{DS}	Resistência de condução do MOSFET.
$R_{SNUBBER}$	Resistor de proteção para o driver L6203.
R_b	Resultante de esforços no ponto (b).
R_a	Resultante de esforços no ponto (a).
$\sum F_y$	Somatória de forças no eixo y.
T_{arc}	Temperatura do arco elétrico.
T_{amb}	Temperatura do ambiente.
T	Tempo de amostragem do ADC.
$T_{setting}$	Tempo de estabelecimento do motor de CC.
t_{ON}	Tempo de acendido do transistor MOSFET.
T_s	Tempo de estabelecimento da função de transferência.
t_p	Tempo de resposta do sensor de velocidade do robô.
U	Tensão do arco elétrico.
V_a	Tensão da armadura do motor de CC.
V_{DS}	Tensão Drenador – Surtidor do MOSFET.
V_{GS}	Tensão Comporta – Surtidor do MOSFET.
$V_{GS(Thr)}$	Tensão Drenador – Surtidor umbral do MOSFET.
V_{BE}	Tensão base emissor do BJT.

V_{SAT}	Tensão de saturação do BJT.
V_C	Tensão do capacitor.
V_f	Tensão de carga do capacitor.
V_S	Tensão da fonte de alimentação.
V_{sense}	Tensão do sensor de corrente do motor de CC.
T_{servo}	Torque fornecido pelo servomotor.
T_f	Torque pelo atrito do eixo do motor de CC.
ΔP	Variância da posição do bico de contato em relação com as peças.
a_0	Valor médio dos coeficientes da série de Fourier trigonométrica.
Aux_n	Variáveis auxiliares para cálculos intermediários dentro do μC .
V_{el}	Velocidade de propagação do som no ar livre.
V_r	Velocidade do rolamento.
$v(t)$	Velocidade linear do robô na direção do cordão de solda.
ω	Velocidade angular do mecanismo da direção.
n	Velocidade do motor de CC em revoluções por minuto.

Sumário

Capítulo 1. Introdução	1
1.1 Descrição do Problema	2
1.2 Objetivos da pesquisa	3
1.3 Organização da dissertação.....	5
Capítulo 2. Revisão Bibliográfica.....	7
2.1 Resenhas de pesquisas sobre sistemas robotizados para soldagem	7
2.2 Variáveis que influenciam o processo de soldagem	16
2.2.1 Tensão e corrente de soldagem	17
2.2.2 Comprimento do arco elétrico e DBCP	21
2.3 Distorções na soldagem	23
2.4 Sensoriamentos na soldagem	25
2.4.1 Sensoriamentos por ultrassom.....	26
2.4.2 Sensoriamentos por infravermelho.....	26
2.5 Fontes de Soldagem MIG/MAG.....	27
Capítulo 3. Materiais, Métodos e Definições Preliminares.....	29
3.1 Introdução	29
3.2 Materiais	31
3.3 Metodologia	33
Capítulo 4. Projeto do Sistema Mecânico.....	35
4.1 Dimensionamento do sistema mecânico.....	35
4.2 Mecanismo de velocidade constante.....	37
4.3 Equações para a análise da resistência à fadiga mecânica.....	43
4.4 Mecanismo da direção	44
4.4.1 Cálculos das dimensões do mecanismo da direção.....	47
4.5 Dimensionamento mecanismo angular.....	51
Capítulo 5. Projeto do Sistema Eletrônico	56
5.1 Definições preliminares para projetar o sistema de controle.....	56
5.2 Sistema de controle geral	62
5.2.1 Bloco sensoriamento da distância (Variação da trajetória).....	62
5.2.2 Bloco atuador (corretor da trajetória).....	64
5.2.3 Bloco Controle da Velocidade Constante	69

5.2.3.1 Modelagem do motor de corrente contínua para o controle da velocidade.....	69
5.2.3.2 Constante contraeletromotriz.....	70
5.2.3.3 Resistência da armadura	70
5.2.3.4 Indutância da armadura.....	71
5.2.3.5 Constante de tempo elétrica.....	72
5.2.3.6 Constante do torque do motor.....	72
5.2.3.7 Constante de tempo mecânica	72
5.2.3.8 Momento de inércia do motor CC.	73
5.2.3.9 Atrito do rotor no motor	73
5.2.3.10 Constante de Viscosidade de Coulomb	73
5.2.3.11 Função de transferência do motor CC	74
5.2.3.11 Sistema com atraso devido ao sensor de velocidade	76
5.2.3.12 Estratégia do controle do motor de CC sem atraso na medição	79
5.2.3.13 Filtro de controle PI.....	80
5.2.3.14 Compensador em tempo discreto.....	82
5.2.3.15 Estratégia de controle para o sistema em laço fechado com o atraso no sensor	84
5.2.3.16 Análise do sinal PWM no motor de CC	87
5.2.3.17 Projeto do gerador do sinal PWM para o motor	91
5.3.2.18 Circuito projetado para regulação do PWM	97
5.2.4 Bloco Sistema de posicionamento CNC	98
5.2.4.1 Obtenção das equações geométricas de posicionamento.....	98
5.2.5 Projeto do sistema de controle dos motores de passo para o sistema linear e o sistema angular	106
5.2.5.3 Bloco Driver Bipolar e controle da corrente.....	112
5.2.5.4 Bloco Sistema Gerador dos Passos.....	116
5.2.6 Sistema de comunicação entre o sistema de posicionamento CNC e o sistema de correção e controle da velocidade constante	119

5.2.7 Bloco Unidade de Processamento Central (UPC).....	121
Capítulo 6. Produtos Mecânicos e Resultados Experimentais dos Sistemas.....	128
6.1 Produtos mecânicos da direção.....	128
6.2 Produto mecânico carrinho	130
6.3 Produto mecânico completo	132
6.4 Funcionamento dos Servos-motores no sistema da direção.	132
6.5 Medições da velocidade do motor aplicando o PWM criado.	133
6.6 Medição e compensação da corrente da armadura com o microcontrolador.....	137
6.7 Comprovação da precisão dos passos do mecanismo angular.....	144
6.8 Comprovação do ângulo final versus a distância medida à peça.....	146
6.9 Comprovação da precisão dos passos do mecanismo linear	148
6.10 Teste do sistema no processo de soldagem MIG/MAG	150
Capítulo 7. Análises de Resultados e Aportes Científicos.....	151
Capítulo 8. Conclusões.	154
Capítulo 9. Sugestões para Trabalhos Futuros.....	155
Referências	156
APÊNDICE A	162
A.1 Código do Microcontrolador PIC16F877A para o Controle da Velocidade e Correção da Trajetória de Soldagem.	163
A.2 Código do Microcontrolador PIC18F4550 para o Posicionamento da Tocha de Soldagem	186
A.3 Código do Microcontrolador PIC10F322 para o Controle da Direção.....	209
APÊNDICE B.....	212
B.1 Código de MatLab™ Programa para o Projeto Mecânico	213
B.2 Código de MatLab™ para a Transformada de Fourier	220

Capítulo 1. Introdução

O presente trabalho apresenta o completo projeto e desenvolvimento de um sistema mecatrônico de maneira integral (solução mecânica e eletrônica) para resolver um problema específico na área de engenharia de soldagem automatizado.

Primeiramente foi definida a concepção do mecanismo que iria a satisfazer as ações necessárias do sistema mecânico para a soldagem de duas chapas de aço com junta em T, processo de alta repetibilidade nos ambientes de estaleiros e fabricação de navios na soldagem de painéis, como se verá nos estudos preliminares na revisão bibliográfica.

O projeto foi baseado nas pesquisas dos sistemas de soldagem automatizados acadêmicos já desenvolvidos, com o objetivo de criar um sistema útil e simples, automatizando o processo de soldagem de juntas em configuração T, evitando o projeto dos sistemas robotizados complexos que tentam resolver com apenas uma única estrutura mecânica (sistemas antropomorfos), todos os possíveis processos de soldagem incrementando a complexidade na programação e configuração das trajetórias nos processos de soldagem.

Uma característica importante do projeto foi a fabricação da maior quantidade de componentes tanto mecânicos quanto eletrônicos, que garantem a independência do projeto a subsistemas criados para outros propósitos, como são as placas dos sistemas CNC (Comando Numérico Computadorizado) comerciais ou sistemas embarcados comerciais criados para aplicações eletrônicas de baixo custo (Arduino), cujas limitações não garantirão o correto funcionamento na nossa aplicação, porque como é sabido a dependência de subsistemas para outros propósitos limitam futuros projetos e dificultam a implantação pelas inúmeras mudanças a serem feitas o que muitas vezes torna-se muito mais complexo.

Procurou-se uma abordagem simples na explicação de cada subsistema ou componente devido à interdisciplinaridade do trabalho que engloba conhecimento de diferentes ramos da engenharia de software, mecânica e eletrônica integrando os saberes para obter um sistema mecatrônico funcional. Foi necessário conhecer o comportamento do processo de soldagem escolhido que é o MIG/MAG, e a influência dos parâmetros de soldagem para a definição das variáveis que permitirão a implementação do sistema.

Para a seleção dos componentes do sistema aplicou-se a metodologia de projeto CAD (Computer Assisted Design); na qual se abrangem conceitos fundamentais de desenvolvimento de mecanismos otimizando a fase de projeto e implementação.

Para conseguir incorporar essa metodologia e melhorar os prazos de projeto e implantação, foi fundamental adquirir uma visão geral do processo de soldagem principalmente quanto às variáveis envolvidas neste processo afetariam o funcionamento dos sensores no processo de manutenção do arco elétrico, além de considerar a disponibilidade dos componentes e peças a adquirir para o projeto. Não menos importante o estudo da importância da automação de processos industriais de alta repetibilidade, maximizando a produtividade e qualidade dos processos de fabricação mecânica, diminuindo os erros humanos, muitas vezes inseridos pelos operários e imperceptíveis ao longo do processo de soldagem, apenas podem ser encontrados após múltiplos ensaios mecânicos das peças finais, aumentando exponencialmente o tempo na produção.

1.1 Descrição do Problema

Geralmente os sistemas automáticos de soldagem são estáticos (sistemas robotizados) limitados a espaços predefinidos para garantir uma boa qualidade nos processos de soldagem e reduzir a complexidade da programação. No entanto, os sistemas robóticos de soldagem são treinados por um especialista em soldagem e este a sua vez replica através de comandos pré-estabelecidos as operações programadas, ou seja, regularmente não há um controle da posição baseado em sensoriamento do posicionamento das peças a serem soldadas. Se as peças sofrem algum deslocamento devido a distorções o robô não agiria e continuaria executando os comandos como foram programados.

Os robôs para soldagem robotizados são regularmente pesados e devem se posicionar e operar em áreas definidas como enxergado na Fig. 1.1.1. O sistema robotizado apresenta muitas vezes uma programação complexa cujo operário novo requer de um treinamento de várias semanas até dominar o funcionamento do sistema o que faz o operário um trabalhador qualificado incrementando assim a complexidade e custo do processo de fabricação.

Figura 1.1.1. Sistema robotizado de soldagem de navios.



Fonte: Chambers (ABB, 2014).

O sistema a se desenvolver, terá uma estrutura que vai sustentar uma tocha de soldagem do tipo MIG/MAG e concomitantemente deslocará e afastará a mesma para manter o arco elétrico da soldagem (e correspondentemente obter a união de duas placas metálicas pelo processo de fusão metálica).

Em princípio, vai se implementar as etapas de afastamento (ou aproximação) da tocha à peça, a variação do ângulo de soldagem e o controle da velocidade da soldagem, é dizer três graus de liberdade, para garantir a maior estabilidade do arco elétrico e interferir minimamente nas variáveis de corrente, tensão, velocidade de soldagem e comprimento do arco elétrico.

Deve se gerar um modelo e concluir na implantação de um laço de realimentação, permitindo a automação do processo de manutenção do arco elétrico.

1.2 Objetivos da pesquisa

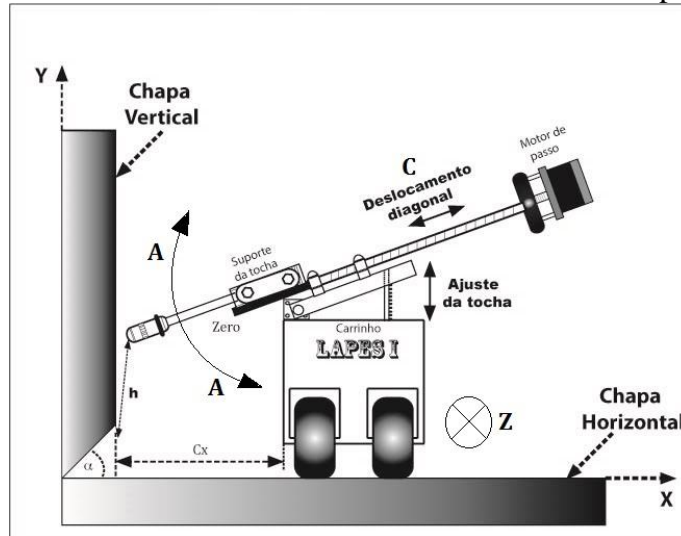
Objetivo geral

Desenvolver um sistema automatizado para o correto posicionamento da tocha MIG/MAG ao longo do processo de soldagem de juntas do tipo T.

Objetivos específicos

- Projetar um sistema mecânico com três graus de liberdade A, C e Z para a movimentação da tocha de soldagem como se mostra na Fig. 1.2.1, baseada na norma NBR 7239 (1982) e WEMAN (2003);

Figura 1.2.1. Graus de liberdade do sistema mecânico a projetar.



Fonte: Próprio autor.

- Projeto dos subsistemas eletrônicos para controle e aquisição dos parâmetros do processo de manutenção do arco elétrico da tocha MIG (Metal Inert Gas);
- Modelagem do sistema mecânico para geração do algoritmo de posicionamento e controle da velocidade;
- Projeto e implantação do algoritmo de controle dos sistemas de manutenção da posição e controle da velocidade de soldagem;
- Montagem do protótipo do sistema para a manutenção do arco elétrico no processo de soldagem;

1.3 Organização da dissertação

No Capítulo 1 é apresentada a preparação da pesquisa, isto é, descrição das principais limitações encontradas nos sistemas robóticos atuais e que podem ser melhoradas. São apresentados detalhadamente os objetivos completos da pesquisa, incluindo também a descrição da organização do documento em questão.

No Capítulo 2, resumem-se os principais tópicos teóricos considerados necessários para desenvolver a pesquisa, entre eles tem-se, uma resenha das pesquisas dos sistemas robotizados para soldagem desenvolvidos na atualidade, sensoriamento por infravermelho, sensoriamento por ultrassom, análise de sinais luminosos infravermelhos nos processos de soldagem, a análise de ondas ultrassônicas na soldagem, a influência das variáveis do processo de soldagem, como corrente, tensão, comprimento do arco e velocidade de soldagem.

No Capítulo 3, apresenta-se uma análise preliminar das variáveis e parâmetros necessários para o projeto mecatrônico descrevendo a metodologia da pesquisa.

No Capítulo 4, foi desenvolvido o projeto do sistema mecânico interligado com o sistema eletrônico, com referencial bibliográfico de autores de projeto de mecanismos.

No Capítulo 5, foi exposto o projeto completo do sistema eletrônico baseado na planta mecânica projetada, com referencial bibliográfico da teoria do controle, funcionamento dos motores elétricos de corrente contínua (CC), e dos sistemas embarcados.

Os Capítulos 4 e 5 são recursivos, pela natureza do trabalho mecatrônico, foi necessário re-projetar várias vezes os sistemas e mecanismos baseado nos resultados de ambos os subsistemas.

No Capítulo 6, mostram-se os resultados do funcionamento do sistema mecatrônico baseado nos algoritmos de controle programados e nas funções matemáticas de posicionamento encontradas.

No Capítulo 7 localiza-se a análises dos resultados da dissertação, além de uma breve descrição dos aportes científicos gerados.

No Capítulo 8 são apresentadas as conclusões do trabalho.

No Capítulo 9 resumem-se sugestões consideradas interessantes para futuras pesquisas baseadas no robô criado.

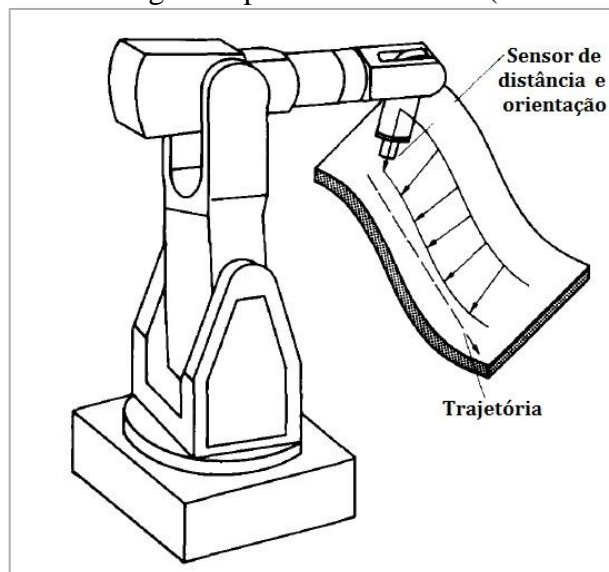
Logo após do Capítulo 9 encontra-se a secção de referências bibliográficas, seguidamente os apêndices do trabalho que incluem, os programas computacionais gerados para o controle dos sistemas e para o processamento da informação.

Capítulo 2. Revisão Bibliográfica

2.1 Resenhas de pesquisas sobre sistemas robotizados para soldagem

A resenha dos sistemas robotizados do interesse desta dissertação começa no ano de 1989, com o sistema auto-programável guiado por sensoriamento segundo Pritschow (1989). O sistema desenvolvido utilizava sensores de distância para auto gerar a programação dos movimentos do robô na trajetória de produção do cordão de solda. Segundo o Pritschow a forma convencional de programação do robô de soldagem, é feita pelo operário quem inseri manualmente os pontos geométricos da trajetória de soldagem, por exemplo, o início e o fim do cordão de solda, através de comandos específicos baseado na geometria das peças metálicas.

Figura 2.1.1. Robô guiado por sensoriamento (varredura da face).



Fonte: (PRITSCHOW, 1989, p.174).

Com ajuda da Fig. 2.1.1, explica-se o conceito geral do sistema do Pritschow, o robô devia ser instruído antes de qualquer atividade, através de um sensor de não contato, mas sempre devia ser fornecido o comando indicativo da trajetória, ou seja, o ponto de início e fim.

O controlador dos seis eixos do robô do trabalho do Pritschow (1989) requeria de um tratamento da informação do sensor para a linearização das coordenadas através do mapeamento inverso, pelo fato da ligação entre o sensor e os eixos coordenados do robô são não lineares. O tipo de sensor utilizado nesta aplicação para a aprendizagem da geometria da peça foram sensores indutivos de proximidade.

A Fig. 2.1.2 mostra um exemplo do tipo de peça que o sistema de Pritschow (1989) podia varrer para digitalizar a face na aprendizagem e logo prosseguir com o processo de soldagem.

Figura 2.1.2. Programação automatizada (testes na face) numa peça onde a superfície deve ser refeita com soldagem.

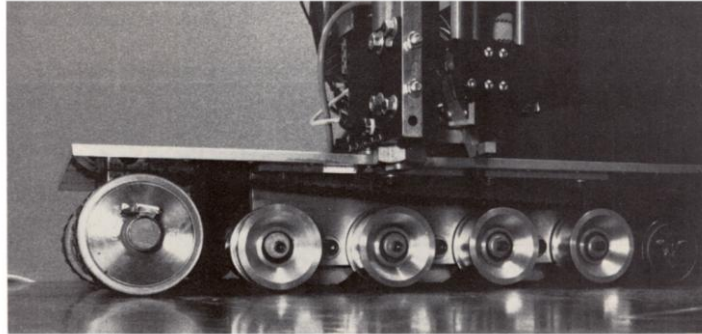


Fonte: (PRITSCHOW, 1989, p.181).

O processo de reconhecimento da geometria desta peça começa marcando os cantos da região de interesse fornecidos pelo operário, após inserir essa informação o sistema grava automaticamente (auto-programável) por varredura o volume da peça a ser soldada. Segundo Pritschow (1989, p. 181), o processo manual para a soldagem deste tipo de peças é de 72 horas, e com o sistema automatizado reduz-se para a metade.

No ano de 1990 Slocum, criou um sistema mecatrônico completo para a soldagem de pinos metálicos, o qual se apresenta na Fig. 2.1.3.

Figura 2.1.3. Sistema mecatrônico para soldagem de pinos metálicos.



Fonte: (SLOCUM, 1990, p. 373).

Dentro das principais características do sistema feito pelo Slocum (1990), têm-se:

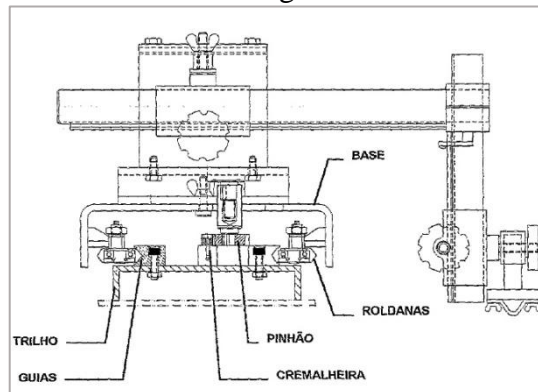
- Automação de processo de alta repetibilidade em estaleiros.
- Sistema adaptável para montagem em trilhos.
- Controlado por computador.
- Atuadores pneumáticos e reles mecânicos.
- Deve ser posta no ponto de soldagem pelo operário e a sua vez corrigir erros na trajetória (dispensa de sensoriamento neste procedimento).
- Possui um sensor indutivo para detecção da superfície embaixo e assim evitar uma queda em altura.

No projeto deste mecanismo, foram considerados aspectos como, cartuchos de pinos a carregar, peso final do sistema a poder ser carregado pelo operário, sistema amigável e simples de usar para evitar a obsolescência do operador e por último a rota a percorrer com os possíveis obstáculos e superfícies do processo (SLOCUM, 1990).

O próximo na resenha é o sistema mecatrônico para soldagem multipasses monitorado por visão computadorizada (FERREIRA, 1997) visto na Fig. 2.1.4. Neste trabalho foi desenvolvido um sistema de monitoramento (principalmente o sistema eletrônico) baseado numa estrutura mecânica existente, para melhorar o processo de soldagem multipasses e reduzir a intervenção do operário na correção da altura da tocha em cada passe da soldagem.

Embora o processo escolhido fosse o MIG/MAG, por fatores como o baixo custo e que é um processo mais limpo, por dispensar da intervenção humana na remoção da escoria, isto facilitaria a implementação do sistema (FERREIRA, 1997, p. 16 - 18).

Figura 2.1.4. Unidade mecânica proposta para o sistema de monitoramento de multipasses de soldagem.

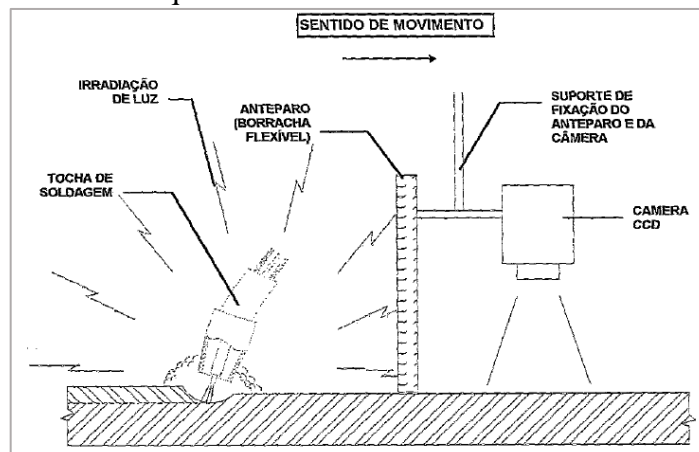


Fonte: (FERREIRA, 1997, p. 38).

O trabalho de Ferreira baseou-se no desenvolvimento de um sistema de visão computadorizada, de imagens do tipo BMP, dado que o sistema mecânico utilizado foi um sistema existente modificado para a aplicação em questão, baseado num arranjo de trilhos para os deslocamentos necessários, mas o conceito adotado foi mais qualitativo. A Fig. 2.1.4, apresenta o mecanismo proposto por Ferreira (1997), no qual não foi feita uma análise dos esforços necessários o dos comprimentos dos trilhos nem das limitações mecânicas nos deslocamentos do sistema, esses parâmetros eram desconhecidos.

O sistema de controle utilizado por Ferreira (1997) consistiu numa placa com relés eletromecânicos se ativando em frequência média, tendo um computador gerando os pulsos de ativação para o controle, no entanto este tipo de sistemas como é bem conhecido produz uma alta Interferência Eletromagnética (IEM) em todos os sistemas elétricos adjacentes, não recomendados para sistemas eletrônicos trabalhando em frequências média-altas.

Figura 2.1.5. Esquemático do sistema de visão com o anteparo.



Fonte: (FERREIRA, 1997, p. 65).

Na implantação do sistema de visão foi colocado um anteparo de borracha para evitar que a luz do arco elétrico saturasse o sensor da câmara segundo a Fig. 2.1.5. Essa configuração não inclui a correção da trajetória em nenhum momento, o que foi feito pelo trabalho de Pritschow (1989), mas nenhum dos dois trabalhos considera a mudança da trajetória no caso acontecer alguma distorção das peças. Cabe a ressalva que o algoritmo de controle do sistema de Ferreira (1997), foi obtido experimentalmente e não teve modelagem alguma para a geração das trajetórias de soldagem em cada passe e sempre precisou de linhas de referência para todas as peças riscadas pelo operário, dado que o sistema de visão somente podia tomar imagens em duas dimensões.

Então todas as distâncias calculadas para os eixos X-Y eram interpoladas em relação com as linhas de referência que sempre eram diferentes, o que faz o cálculo do erro do sistema no posicionamento maior com cada passe, alterando significativamente os parâmetros da soldagem. Um ponto fraco detectado no trabalho de Ferreira, é que o cálculo de avanço dos eixos é feito baseado em variáveis temporais, interrompendo o motor após um tempo proporcional à distância, desconsiderando os atrasos temporais da comunicação, o cálculo do computador, o tempo de resposta mecânico dos reles e o tempo de parada pelo momento de inércia no movimento do motor (não existiu um algoritmo de frenagem), o que aumenta ainda mais o erro no posicionamento e deslocamento.

Em 2002 foi desenvolvido o trabalho de Gohr (2002), que baseado nas produções de outros autores, criou uma série de algoritmos de controle computacionais aprimorando as variáveis de tensão e corrente da soldagem, para melhorar as técnicas existentes comumente utilizadas em processos de soldagem no Brasil. Neste trabalho não foi desenvolvido um sistema mecânico, no entanto apresentam diversas técnicas de soldagem no processo MIG/MAG permitindo observar um melhor entendimento do processo, por exemplo, como age a corrente e a tensão desse processo em situações diferentes.

Na pesquisa de Gohr (2002) foi interessante observar os métodos criados para a reignição do arco elétrico tendo diferentes valores de corrente e da DBCP (distância do bico de contato a peça). Nenhum dos “projetos” eletrônicos de potência foi explicado, nem se fez referência a algum autor de sistemas de potência.

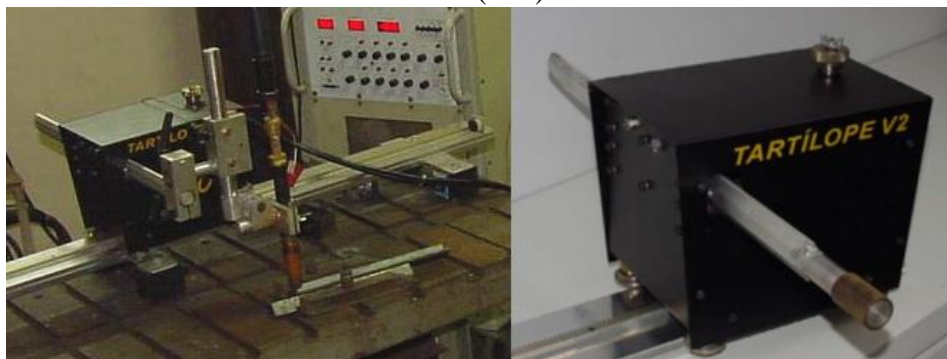
A automação do processo de soldagem GMAW (Gas Metal Arc Welding) foi investigado por KIM et al. (2003), utilizando os parâmetros de soldagem de corrente entre os valores de 170 - 270A, tensões entre os valores de 23 - 28V, velocidade de soldagem entre os

valores de 12 – 50 cm/min e o material escolhido foi o aço BV-AH 32, e na técnica de soldagem de uma junta de topo multipasses numa chapa de 12 mm de espessura e o gás de proteção utilizado foi o Ar (80%) + CO₂ (20%). Para a análise, foi utilizado um modelo baseado em redes neurais (KIM et al. 2003, p.570) onde foi determinado que todos esses parâmetros influenciam no tamanho do reforço em cada passe com a soldagem robotizada, foi determinado que o modelo baseado nestas redes neurais podem ajudar a prever os parâmetros de corrente, tensão e velocidade da soldagem a se utilizar na soldagem robotizada. Esses parâmetros utilizados por Kim poderão ser tomados como referência no desenvolvimento do sistema mecatrônico do presente projeto.

Continuando com a resenha dos sistemas mecatrônicos, no ano de 2003, no trabalho de Costa (2003), feito no Laboratório de Pesquisa de Soldagem da Universidade Federal de Santa Catarina (LABSOLDA), foi desenvolvido um sistema de controle para ser integrado em um sistema robotizado existente, dito controle utilizou a corrente e a tensão do arco elétrico para seguir a junta no processo de soldagem. Nesta pesquisa foi utilizado um sistema existente chamado “Tartílope V2”, que consistia de um mecanismo de trilhos adaptáveis, já modificado de uma versão anterior. O trabalho em menção faz um resumo dos diferentes tipos de sensores utilizados para a criação de sistemas de seguimento de junta (COSTA, 2003 p. 6 - 9), fazendo destaque do sensor de seguimento de junta com ajuda das variáveis do arco.

Lembrando que o principal intuito foi desenvolver o sistema de seguimento se baseando nas variáveis do arco, já tendo uma grande infraestrutura mecânica pronta que precisou de pouquíssimas modificações. Outro detalhe deste trabalho é que o sistema de detecção apenas pode operar aplicando o movimento de tecimento no processo de soldagem. O sistema “Tartílope” utilizado neste trabalho apresenta-se na Fig. 2.1.6.

Figura 2.1.6. Sistema Tartílope V2.0 utilizado para o projeto do Sistema de Seguimento de Junta (SSJ).



Fonte: (COSTA, 2003, p. 26 e 30).

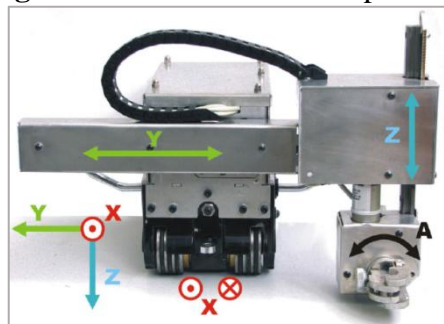
Uma continuação do projeto “Tartílope V2.0” foi apresentada dois anos mais tarde (BROERING, 2005), tendo nesta versão melhoras na programação e uma maior robustez mecânica, para ser utilizado em condições mais severas.

Neste trabalho Broering faz uma conjunção de trabalhos anteriores, integrando ao sistema Tartílope o sensor desenvolvido por Costa (2003) e integrando uma nova estrutura mecânica para soldagem orbital, que segundo o mesmo Broering foi desenvolvido por outros membros da sua equipe de trabalho. Principalmente, o grosso do trabalho de Broering foi otimizar o sistema Tartílope para ser um dispositivo mais “robusto”. Os drivers de potência, as placas de controle e processamento foram fabricadas por terceiros, e o sistema de controle foi mudado para um computador do tipo industrial, o driver do motor de CC (corrente contínua) foi comprado de terceiros.

No quinto Congresso de Engenharia Mecânica (CONEM) no ano de 2008, foi apresentada uma nova versão do projeto Tartílope chamada “Tartílope V4.0”, o qual apresentou uma nova estrutura mecânica robusta e um novo sistema de controle integrando agora um sistema de posicionamento numérico chamado de “MACH 3”, que é um programa do tipo “software livre” para interpretação de coordenadas provenientes de arquivos em formatos CAD (STEINBACH, 2008). Essa mudança no “Tartílope” faz o sistema apenas aproveitável em pesquisas, devido às condições de usabilidade do software livre, além de elevar a complexidade do uso do equipamento a pessoas altamente qualificadas (usabilidade de ferramentas para projeto CAD).

Cabe ressaltar que essa não é a premissa da automação, que pelo contrário, busca facilitar o uso dos equipamentos e retirar da equação a dependência dos operários qualificados nos processos de produção de alta repetibilidade. Uma imagem da versão apresentada por Steinbach (2008) é vista na Fig. 2.1.7.

Figura 2.1.7. Sistema Tartílope V4.0.



Fonte: (STEINBACH, 2008, p. 2).

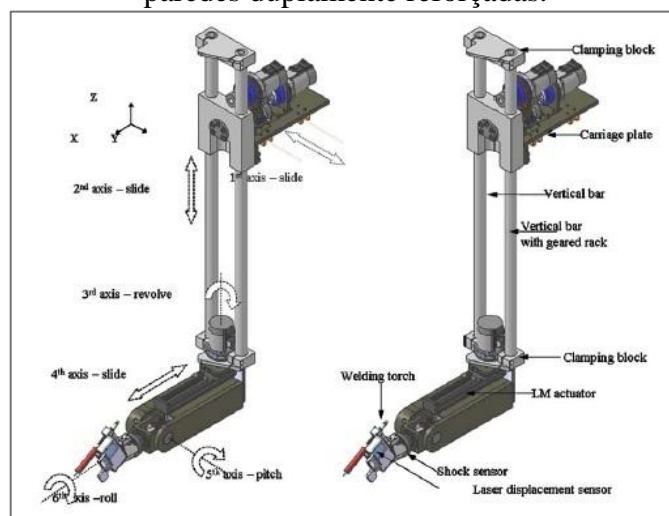
Nessa versão da Fig. 2.1.7, o Tartilope V4.0, apresenta um grau de liberdade a mais em relação com a versão anterior que é o eixo Z e rodas magnéticas para aplicações de soldagem em dutos ou tubos de caldeiras, no entanto os ensaios demonstraram que essas rodas não eram funcionais 100% (STEINBACH, 2009) e foram mudadas para rodas de borracha. Interessante enxergar como o sistema Tartilope gradativamente faz-se dependente de tecnologias feitas por terceiros o que resta real flexibilidade e potencial de desenvolvimento próprio.

No ano de 2010, Lee (2010) concebeu um sistema robótico de soldagem com seis graus de liberdade para a soldagem de navios com o casco duplamente reforçado conforme a Fig. 2.1.8, pesquisa patrocinada pela *DAEWOO Shipbuilding* da Coreia. O mecanismo foi projetado para utilizar a mesma estrutura do navio como trilhos.

Segundo Lee (2010), a função principal do robô era fazer a soldagem das bordas na forma U dos blocos fechados dos navios. A altura do robô de Lee era de 1,4 m e o comprimento de 1,8 m, suficiente para gerar os deslocamentos necessários do tipo de soldagem requerida.

O peso total do robô era de 250 kg, excluindo a tocha e o rolo do arame. O dispositivo mecânico criado por Lee possui um sensor de distância a laser acoplado no suporte da tocha para o seguimento da junta (sistema auto-programável fora de linha), além de um sensor de choque no caso o sistema possa bater na peça e por fim um sensor do arco elétrico do tipo SSJ (sensor de seguimento de junta).

Figura 2.1.8. Sistema robótico com seis graus de liberdade para soldagem de navios com paredes duplamente reforçadas.

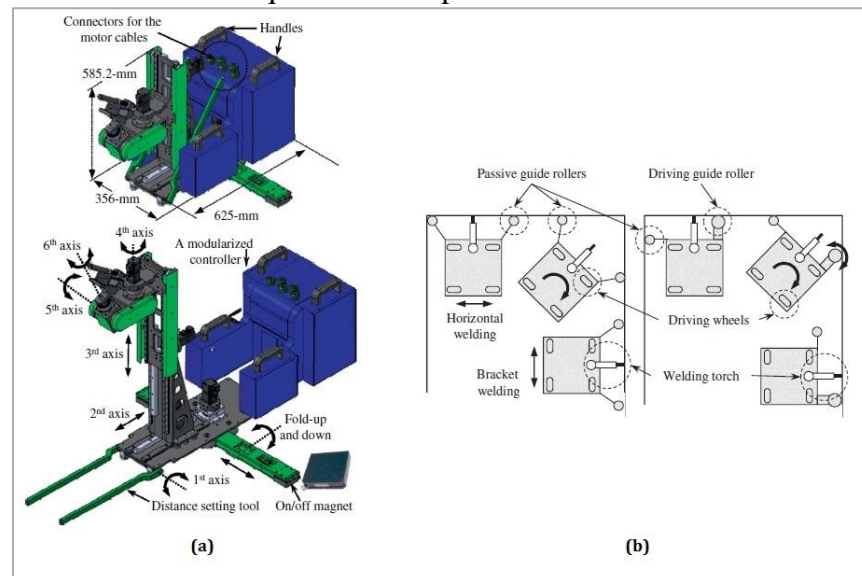


Fonte: (LEE, 2010, p. 16).

Lee et al. (2011), desenvolveu um novo robô inteligente para soldagem de navios, que pudesse ser autônomo para ser utilizado num ambiente hostil para os seres humanos.

O protótipo visto na Fig. 2.1.9, é uma versão modificada do sistema apresentado na Fig. 2.1.8, integrando ao equipamento ferramentas de posicionamento mecânico de contato para identificar o espaço de trabalho do robô e assim melhorar o seguimento do processo na forma U, no entanto este novo produto possui agora rodas para deslocamento livre no solo, deixando de lado os trilhos da versão anterior e melhorando o tempo de aprendizagem na identificação da trajetória de soldagem em relação a versão de 2010, além de integrar um sistema de comunicação sem fio para acelerar o processo de programação.

Figura 2.1.9. Sistema robótico inteligente para soldagem de navios (a) Robô de soldagem (b) Esquemático do processo.



Fonte: (LEE, 2011, p. 381).

Polezi (2012), apresentou uma modificação do “Tartílope V2.0”, melhorando o software dessa versão e colocando novas funções nesse gênero todo isto no CONEM em 2012.

Dez anos após do trabalho de Costa (2003), Kalil (2013) revalidou e aperfeiçoou o sensor de seguimento de junta usando o arco elétrico como variável de feedback. Nessa oportunidade Kalil, utilizou o mesmo sistema mecatrônico o Tartílope V2.0. Importante ressaltar ao fato do que Kalil faz uma análise exaustiva do comportamento do arco com a DBCP, principal ferramenta de diagnóstico e monitoramento do algoritmo de sensoriamento de Costa e que foi otimizada por Kalil.

Pan (2012) fez um estudo dos avanços tecnológicos atingidos no gênero da programação e sensoriamento para robôs (utilizados em processos indústrias), desde 2001 até o 2008, incluindo programação online, online assistida, off-line, guiada por sensoriamento online, entre várias outras. Ressaltando do trabalho de Pan, nos últimos anos os progressos no ramo da programação de robôs têm se focalizado mais no equipamento mesmo, ou seja, fazer com que o sistema tenha infinita gama de funções integradas apenas num dispositivo, esquecendo talvez que a automação deve se focalizar nos processos a automatizar e quais variáveis do processo ou que acontecimentos jazem nele que possam ser integradas no sistema robótico.

Então as pesquisas estão se orientando (excetuando a Kalil 2013) pela otimização dos sistemas e tecnologias existentes e após disso na procura de processos nos quais esses equipamentos possam ser empregados. Ao invés de estudar plenamente o processo industrial e tentar melhorar o mesmo e logo depois automatizá-lo baseado nesse conhecimento, está se esquecendo do processo industrial e os especialistas estão encarando os problemas de robótica sem se importar pelo problema a se resolver.

2.2 Variáveis que influenciam o processo de soldagem

Segundo Modenesi (2011a) a soldagem é um processo de união de dois metais por diversos métodos, afetando e mudando as propriedades interatômicas ou intermoleculares dos metais, empregando o calor para iniciar a deposição de material de solda (quando utilizado) e conseguir a união de duas peças através da fusão do material, criando microestruturas na ZAC (zona afetada pelo calor), com propriedades mecânicas diferentes do material de base inicialmente utilizado.

A energia necessária para atingir a união das peças depende de diversas variáveis, no caso do processo de soldagem por fusão MIG/MAG a maioria dessas está associada ao arco elétrico, fonte de calor que destrói as camadas nas superfícies ao longo da junta para permitir a união das peças. Entre as variáveis importantes temos a tensão e a corrente do arco elétrico, o comprimento do eletrodo, a DBCP (distância do bico de contato a peça), o ângulo do eletrodo.

Essas variáveis alteram significativamente a quantidade de calor a ser transferida do arco elétrico à peça. Porém mantendo essas variáveis constantes ao longo do processo de soldagem, a qualidade e a produtividade do processo de fabricação podem ser melhoradas.

A soldagem é um processo largamente utilizado nos processos de fabricação de navios, com ambientes e condições de trabalho extremas, prejudiciais para os colaboradores humanos (LEE et al., 2011). Estas características do ambiente de trabalho, além da tentativa por melhorar os processos de produção e fabricação de navios, acompanhada com a criação de novos sistemas de soldagem mecatrônicos simples, é que precisa do domínio das áreas de conhecimento envolvidas.

O conhecimento e entendimento das variáveis do processo de soldagem MIG/MAG e como estas afetam conseqüentemente o arco elétrico e, portanto influenciam na transferência do calor às peças, é vital para o desenvolvimento de projetos de automação e soldagem robotizada.

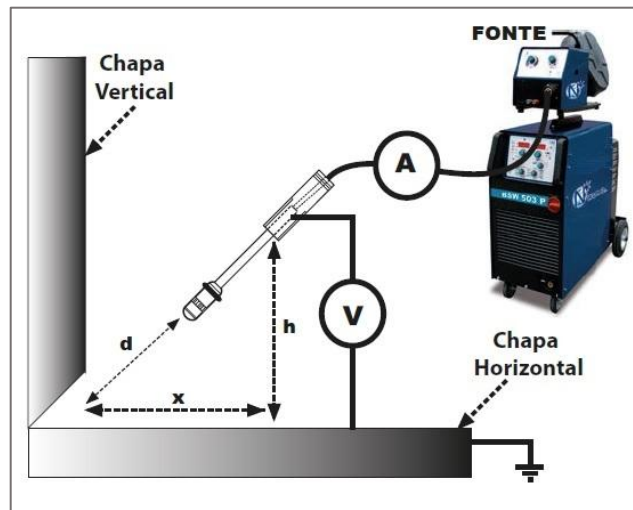
2.2.1 Tensão e corrente de soldagem

A tensão e a corrente elétrica na soldagem especificamente no processo MIG/MAG é surtida pela fonte de energia, umas das principais partes do equipamento.

A amplitude da tensão é regulável, entre 12 e 50 volts, usada em conjunto com o sistema de alimentação de arame; a diferença de potencial medida no arco elétrico é mantida constante pela fonte ao longo do processo de soldagem variando a corrente elétrica por conseqüência da variação da velocidade de alimentação do arame.

A Fig. 2.2.1 mostra as variáveis de tensão e corrente no processo MIG/MAG para uma junta de soldagem em configuração T.

Figura 2.2.1. Circuito fechado de tensão e corrente na soldagem.



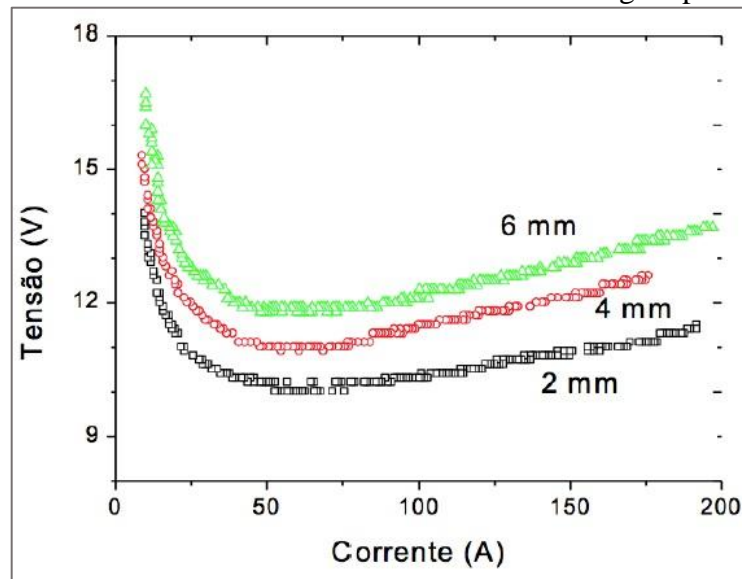
Fonte: Próprio autor

No processo de soldagem MIG/MAG a transferência do metal de adição depende da combinação destas variáveis, associado também as características do eletrodo e a sua velocidade de alimentação. Existem três tipos de transferência do metal de adição nesta técnica, a transferência globular, por spray e a mais comum a transferência por curto-circuito. O nível da tensão, da corrente e da polaridade, diferi de um tipo de transferência para outro, e a estabilidade do arco e a qualidade da solda depende do comportamento e variação das variáveis de corrente e tensão (MODENESI, 2011b).

Modenesi (2011a) na Fig. 2.2.2, mostra claramente que a relação entre a tensão e a corrente no arco elétrico na soldagem não é linear. Isto evidentemente devido ao fato que a diferença de um resistor, no arco elétrico é gerada entre dois eletrodos chamados cátodo e ânodo. A Fig. 2.2.2 mostra o comportamento da tensão para diferentes valores de corrente mantendo constante o comprimento do arco.

Como foi dito, essa especial característica de geração do arco elétrico entre dois eletrodos tendo como meio condutor um gás, produz essa variação exponencial da curva corrente-tensão na soldagem. O potencial de excitação do gás para a abertura do arco deve ser maior que a queda de tensão entre o ânodo - cátodo (V_{AK}).

Figura 2.2.2. Curvas características de tensão e corrente na soldagem para diferentes DBCP.

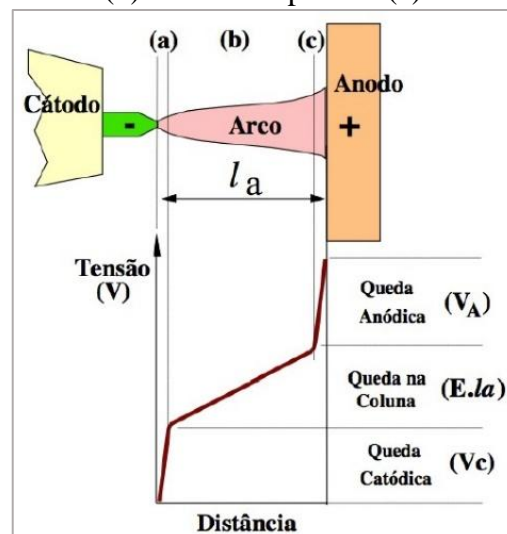


Fonte: (MODENESI, 2011a, p. 5).

Na Fig. 2.2.3 indica-se conceitualmente como é distribuída essa tensão ao longo do arco elétrico tendo que na coluna de plasma o campo elétrico distribuído de CC define o valor da tensão de CC por unidade de comprimento. A Eq. (2.2.1) permite aproximar a curva característica da tensão-corrente para um dispositivo com dois eletrodos do tipo efeito de campo elétrico.

$$I_D = I_S e^{\frac{V}{V_T}} \quad V > V_T \quad (2.2.1)$$

Figura 2.2.3. Potencial elétrico das regiões que conformam o arco elétrico de soldagem (a) cátodo (b) Coluna de plasma (c) ânodo.



Fonte: (MODENESI, 2011a, p. 8).

Na Eq. (2.2.2) o potencial térmico depende do tipo de material e condições de temperatura.

$$V_T = \frac{n_a k T_{arc}}{q} \quad (2.2.2)$$

As expressões das Eq. (2.2.1) e (2.2.2) são largamente conhecidas na área da eletrônica e representam o comportamento eletrônico dos diodos semicondutores, neste caso como visto na Fig. 2.2.3, a configuração é equivalente à de diodos eletrônicos, tendo que a corrente de saturação (I_S) da Eq. (2.2.1), depende da área da coluna de plasma.

Nos processos de soldagem GMAW, ... , as velocidades de alimentação e de fusão do eletrodo, o stickout, o comprimento de arco (a tensão) e a corrente de soldagem estão todos relacionados, não sendo possível alterar uma variável mantendo as outras constantes (MODENESI, 2011a p. 31).

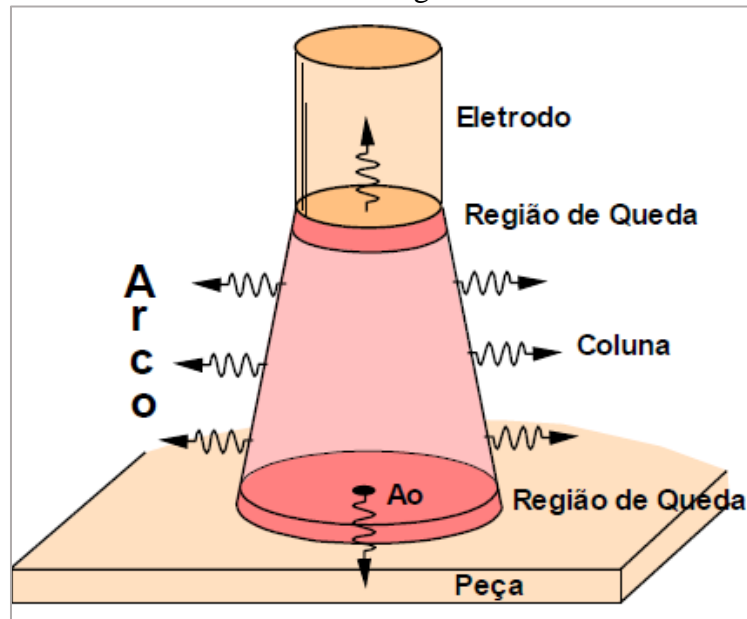
Segundo Modenesi, é incorreto tentar controlar uma dessas variáveis e mudar as outras, o mais lógico a se fazer é manter constantes todos esses parâmetros num processo específico, uma vez sejam fixadas.

A corrente e a tensão no processo de soldagem são importantes em conjunto porque são as responsáveis de gerar o calor que se transfere para a peça a se soldar. Essa potência toma a forma da Eq. (2.2.3).

$$P_{esp} = \frac{\eta UI}{A_0} \quad (2.2.3)$$

Essa área de contato é vista na Fig. 2.2.4, e depende do stickout, se essa distância aumenta, concomitantemente o fará a área de contato.

Figura 2.2.4. Representação simplificada das direções dominantes de fluxo de calor nas diferentes regiões do arco.



Fonte: (MODENESI, 2011a, p.33).

Segundo Modenesi (2011a, p. 35), a corrente e a tensão da soldagem são variáveis primárias no processo de soldagem.

2.2.2 Comprimento do arco elétrico e DBCP

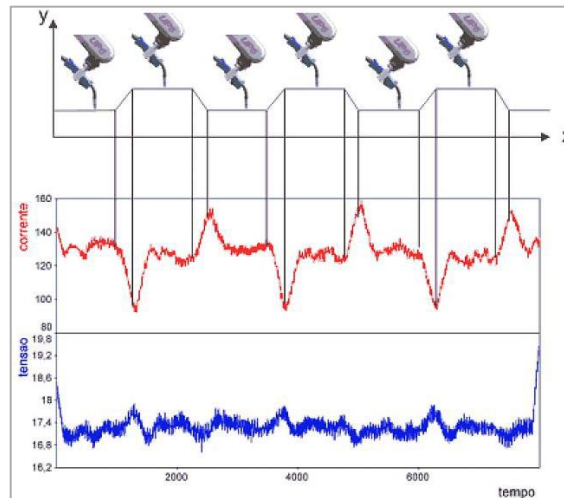
Segundo Nascimento (2001 apud Tong et. al, 2005c, p. 25) um fator que produz a instabilidade do arco elétrico na soldagem MIG é o sistema de controle do comprimento do arco elétrico. “De qualquer forma, o controle do comprimento do arco, sobretudo em polaridade variável é determinante para se obter estabilidade do arco satisfatória” (NASCIMENTO, 2011 p. 25).

Segundo Alves (2009, p. 20), quanto maior for o comprimento do eletrodo (stickout), maior será o aquecimento deste e menor a velocidade de soldagem necessária para a fundição dele. Uma extensão do eletrodo boa é entre 6 – 13 mm, para transferência por curto circuito, e de entre 13 – 25 mm para outros modos.

Em soldas de filete na posição horizontal o eletrodo deve estar posicionado num ângulo de 45 graus entre as peças (ALVES 2009, p. 21). Como foi observado na Fig. 2.2.2,

uma mudança na DBCP automaticamente produzirá uma mudança na corrente e na tensão do arco, isto é confirmado na Fig. 2.2.5.

Figura 2.2.5. Comportamento da corrente e da tensão médias do arco elétrico para alterações da DBCP num ensaio robotizado.

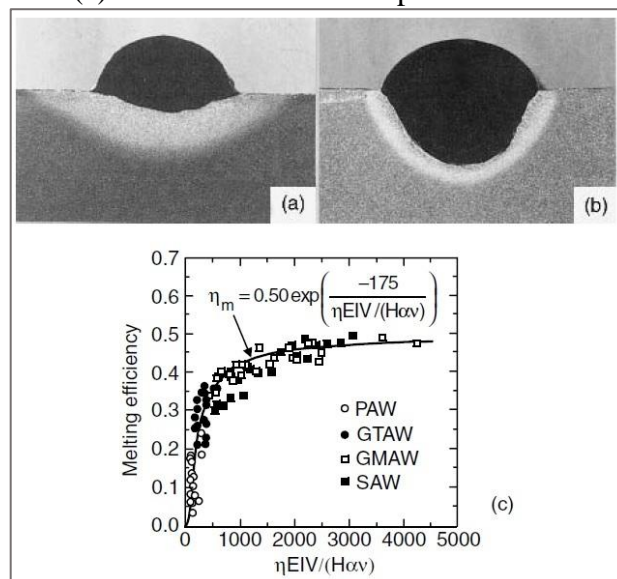


Fonte: (KALIL, 2013, apud Silva 2005, p. 46).

2.2.3 Velocidade da Soldagem

A velocidade de soldagem segundo Kou (2003, p. 45), afeta o rendimento de fusão das peças como visto na Fig. 2.2.6.

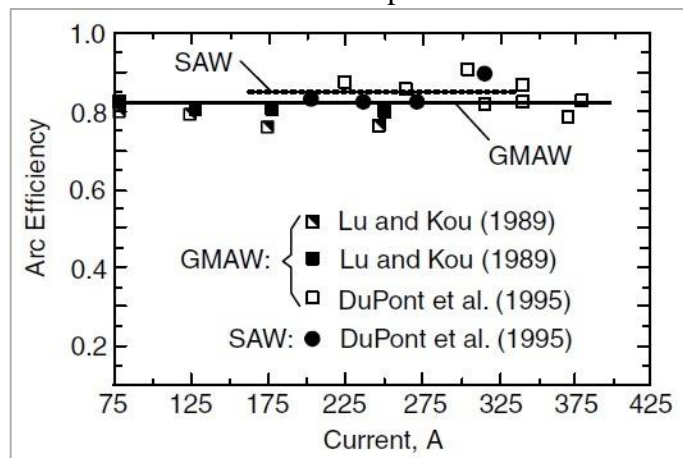
Figura 2.2.6. Rendimento de fusão (a) baixa relação Calor/velocidade de soldagem (b) Alta relação Calor/Velocidade (c) Curva de rendimento-parâmetro adimensional $\eta UIV/(H\alpha v)$



Fonte: (KOU, 2003, p. 44).

Na Fig. 2.2.6, cabe ressaltar que a velocidade de soldagem não pode aumentar, se a potência da fonte não é aumentada. Dito isto então, para uma potência fixa no processo de soldagem deve-se manter fixa a velocidade de soldagem para manter estável o rendimento de fusão. Por outro lado enxerga-se no gráfico da Fig. 2.2.7, que mesmo variando a corrente não é alterado o rendimento térmico, que se mantém próxima a 80% na soldagem GMAW (Gas Metal Arc Welding) e um pouco maior para a soldagem SAW (Submerged Arc Welding), isto para uma mesma velocidade de soldagem.

Figura 2.2.7. Eficiência do arco elétrico dos processos SAW e GMAW(MIG/MAG)



Fonte: (KOU, 2003, p. 40).

2.3 Distorções na soldagem

Na soldagem a arco elétrico ocorre transferência e a condução do calor no material, gerando desta forma, uma distribuição de temperatura não uniforme ao longo de todo o material. Como consequência desta distribuição não uniforme, acontece uma dilatação e contração de forma desigual no material, produzindo assim uma distorção na estrutura e uma acumulação de energia interna denominada de tensão residual.

Sabendo-se disso, alguns autores (KOU, 2003; RADAJ 2003) definem distorção como uma expansão (ou bem a contração) não uniforme do cordão de solda e do metal de base durante o ciclo térmico do processo de soldagem. Modenesi (2001) trata a distorção como uma mudança de forma e dimensões de componentes soldados devido às deformações térmicas do material durante a soldagem. Soldagem em excesso, soldagem em juntas livres, seleção incorreta do chanfro ou da sequência de soldagem, entre outros fatores podem causar

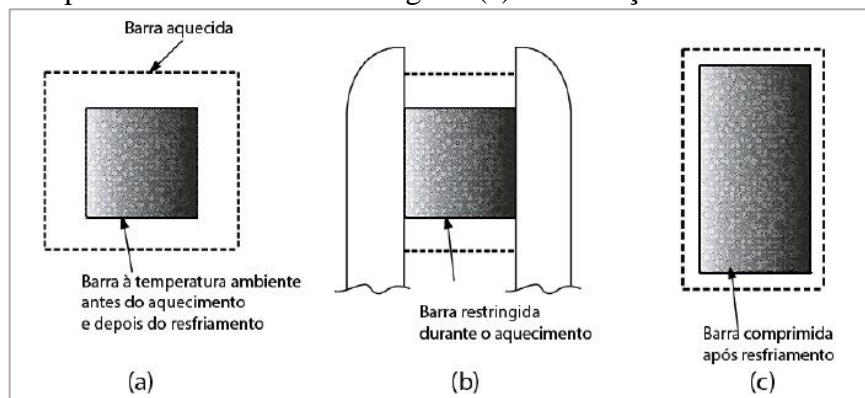
esse problema. No entanto, esse efeito pode ser reduzido utilizando técnicas para isso. A correção da distorção em soldas prontas exige medidas onerosas, como desempenamento mecânico ou térmico, remoção da solda e a re-soldagem.

Para compreender como ocorre a distorção durante o aquecimento e resfriamento do metal de base considere-se a barra de aço mostrada na Fig. 2.3.1.

O processo de aquecimento e resfriamento é mostrado, onde ao final do processo se dá uma deformação da barra pelas restrições forçadas durante o estágio de expansão, esta deformação resultante é uma forma simples de explicar porque ocorrem as distorções.

Em uma junta soldada, essas mesmas forças de expansão e tração agem no metal de solda e no metal de base. Quando o metal de solda solidifica-se e funde-se com o metal de base, este está em sua máxima expansão. Durante o arrefecimento, o metal tenta contrair para o volume que normalmente ocupa a temperatura mais baixa, mas é impedida de fazê-lo pelo metal de base adjacente. Por isso, as tensões se desenvolvem dentro da soldadura e o metal de base adjacente. Neste ponto, o cordão de solda “afina”, ajustando assim as exigências de volume do que a temperatura mais baixa.

Figura 2.3.1. Ocorrência das distorções numa barra de aço (a) aquecendo e resfriando sem restrições físicas (b) restringindo a expansão durante o aquecimento produz uma deformação da barra ao se expandir onde não for restringida. (c) deformação ocorrida na barra ao resfriar

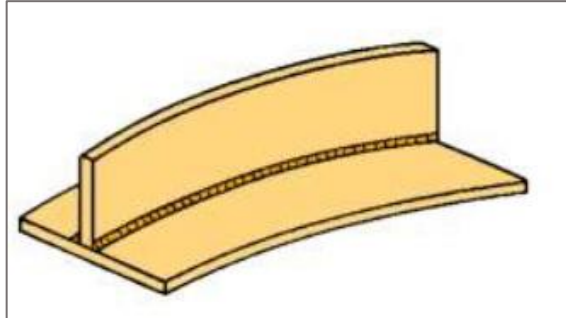


Fonte: Alterada de Lincoln, Electric (2013)

Mas apenas as tensões que excedem o limite de escoamento do metal de solda são aliviadas por este esforço. No momento que a solda atinge a temperatura ambiente assumindo retenção completa do metal de base, de modo que não pode se mover a solda irá conter tensões de tração acumuladas, aproximadamente igual à resistência à deformação do metal. Se os apoios (grampos que seguram a peça de trabalho, ou uma força de tração opostas) são

removidos, as tensões residuais são parcialmente aliviadas causando que o metal de base possa se mover, distorcendo assim o material. Um exemplo destas distorções nas peças soldadas apresenta-se na Fig. 2.3.2.

Figura 2.3.2. Exemplo da distorção numa soldagem de chapas em junta T



Fonte: (MODENESI et al. 2011c, p. 4.19)

2.4 Sensoriamentos na soldagem

O sensoriamento dos parâmetros físicos de um sistema é um dos pilares mais importantes na automação de qualquer processo industrializado. Um sensor utilizado nos processos de soldagem é um instrumento ou dispositivo, que pode fazer parte do mecanismo de soldagem de forma geral (automática ou não) (NAIDU, 2003 p. 95). Os sensores de soldagem podem se qualificar nas categorias a seguir:

Sensores de contato afastados da junta:

- Esfera ou roda de contato.
- Contato elétrico com a peça (pode ser o arame mesmo).

Sensores sem contato afastados da junta:

- Sensores indutivos.
- Sensores capacitivos.
- Ultrassônicos.
- Óticos de posicionamento.
- Óticos de perfil.
- Infravermelhos de varredura.

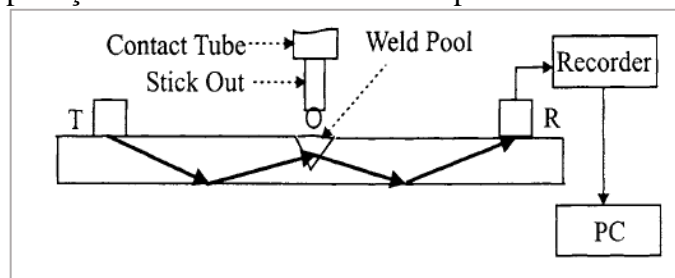
Sensores sem contato próximos à junta:

- Sensores de arco elétrico.
- Sensores de comprimento de arco (óticos).
- Sensores de ressonância da poça (óticos ou elétricos).
- Sensores da poça (linha espectral).
- Sensores de radiação traseira.
- Sensores de geometria da poça.

2.4.1 Sensoriamentos por ultrassom

O uso típico dos sensores por ultrassom na soldagem é representado na Fig. 2.4.1.

Figura 2.4.1. Aplicação do sensor ultrassônico no processo de soldagem MIG/MAG



Fonte: (NAIDU, 2003, p. 107)

Para o caso da aplicação na Fig. 2.4.1, o sensor por ultrassom consta de dois transdutores que são o Transmissor e o Receptor, são colocados nos dois extremos da fonte de calor. Esta aplicação é utilizada principalmente para detecção de defeitos na solda ou para detectar a dimensão da poça de fusão.

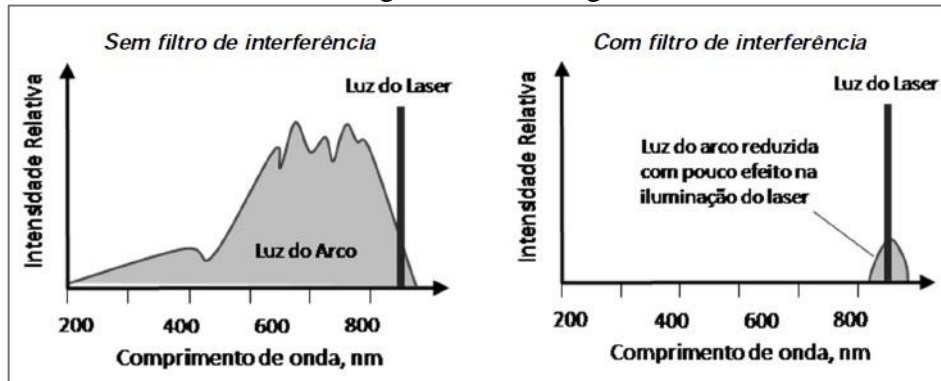
Fathi e Nezih (2011) utilizaram os sensores de ultrassom para medir as tensões residuais que ficam na peça após o processo de soldagem.

2.4.2 Sensoriamentos por infravermelho

O sensoriamento por infravermelho na soldagem MIG/MAG, é largamente utilizado nos sistemas de visão computadorizada. Os sistemas que utilizam esta tecnologia precisam de

filtros ópticos para eliminar a luz de alta potência emanada pelo arco elétrico o que saturaria qualquer sensor com esse comprimento de onda. Esse fenômeno é mostrado na Fig. 2.4.2.

Figura 2.4.2. Medição da intensidade luminosa do arco elétrico na soldagem com filtragem e sem filtragem



Fonte: (PIMENTA, 2011, apud Houghton et al. 2005, p. 9)

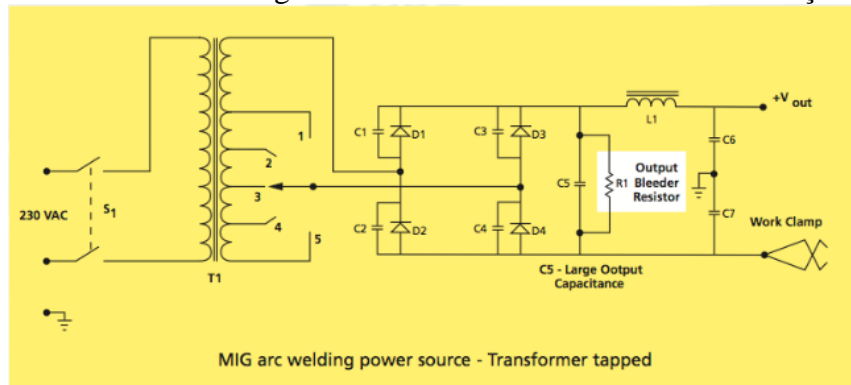
Pelo que qualquer sensor infravermelho apenas poderá ser utilizado off-line do processo de soldagem se não é colocado um filtro de interferência que bloqueie a luz fora do comprimento de onda entre os 800 – 900 nm.

2.5 Fontes de Soldagem MIG/MAG

Na soldagem MIG/MAG, ajusta-se a tensão na fonte, assim, a tensão define o comprimento do arco, resultado da característica de auto ajuste. Se o soldador afasta a tocha, o comprimento do arco não se altera, em vez disso, o comprimento do eletrodo é alterado (“stickout”). A corrente não pode ser ajustada diretamente, depende diretamente da velocidade de alimentação do arame (e o seu diâmetro) usado.

Regularmente as fontes MIG/MAG são de tensão constante (VC) que é aquela que permite a regulagem da tensão de trabalho e que apresenta uma curva característica estática (CEF), que tende a produzir uma tensão de trabalho relativamente constante (neste caso, a corrente de soldagem passa a não ser regulável, tornando-se dependente da fixação dos demais parâmetros). Um circuito típico para uma fonte de soldagem MIG é apresentado na Fig. 2.5.1.

Figura 2.5.1. Fonte da soldagem MIG com transformador com derivação central.



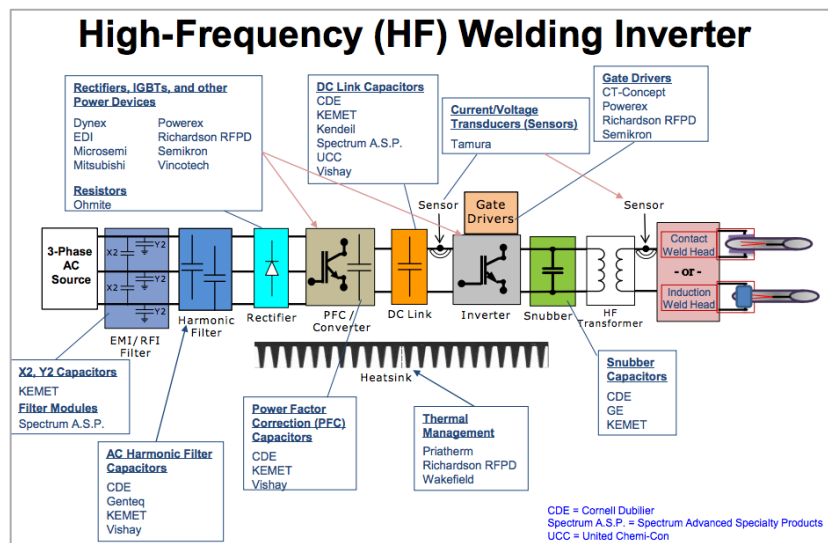
Fonte: TT Electronics (2013).

Fontes mais modernas e com sistemas de controle e regulação mais efetivos são da forma apresentada na Fig. 2.5.2.

Estas fontes apresentam circuitos de controle de alta precisão, baseados no chaveamento em alta frequência com todo tipo de proteções e filtragem para garantir um sinal de corrente e tensão constantes e de boa qualidade. Sistemas como o da Fig. 2.5.2, são projetados para máxima eficiência e rendimento garantindo também a mais alta segurança para os usuários de acordo com as normas internacionais.

Estas fontes já possuem o sistema de controle da velocidade de alimentação do arame, reignição do arco, entre outras funções, o que garanta uma melhora na qualidade da solda.

Figura 2.5.2. Fonte da soldagem de última geração com sensoriamento e controle baseado nas variáveis de tensão e corrente do arco.



Fonte: RichardsonFPD (2013).

Capítulo 3. Materiais, Métodos e Definições Preliminares

3.1 Introdução

Neste capítulo serão definidos os materiais requeridos e a metodologia necessária para resolver o problema em questão.

Logo depois de uma visita ao estaleiro da cidade de Rio Grande, no estado de Rio Grande do Sul, conseguiu-se visualizar a grande necessidade que havia nos estaleiros de equipamento de soldagem que não fosse sofisticado, isto é, um dispositivo simples e prático de utilizar, que amortize ao máximo o tempo de treinamento dos técnicos para o início das atividades de soldagem, todo justificado nos prazos estabelecidos nos contratos que deviam ser cumpridos para a entrega das plataformas.

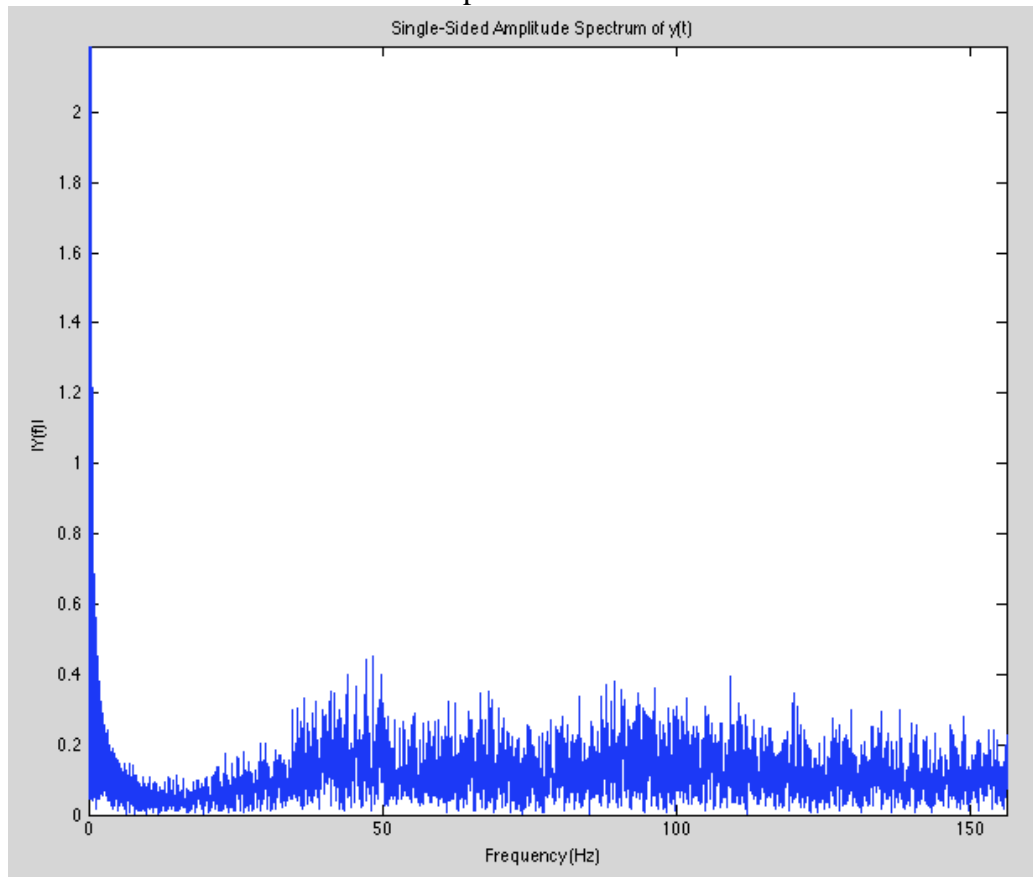
Constatou-se que o tempo de treinamento dos técnicos para o uso de equipamento especializado para soldagem poderia demorar até dois meses, reduzindo em grande medida a produção do estaleiro.

Dito isto, introduz-se na solução buscada a ideologia de fazer um sistema onde o técnico necessite de um menor tempo de treinamento.

Além dessas novas características, pensou-se num sistema leve, que poderia ser carregado pelo soldador, ser deslocado através do estaleiro sem maior dificuldade, e por fim, num sistema livre sem a necessidade de trilhos, guias ou qualquer outro acessório que limite o mecanismo a espaços pré-definidos, resolvendo como foi estabelecida no projeto uma configuração específica no tipo de junta que é a junta em T.

Para entender o comportamento da corrente e tensão da soldagem no processo MIG/MAG foram feitas medições preliminares destas variáveis para diferentes distâncias do bico de contato à peça (DBCP) utilizando o robô MOTOMAN do laboratório de pesquisa em engenharia de soldagem da Universidade Federal de Rio Grande - FURG (LAPES). Para todas as medições do conjunto tensão-corrente da soldagem (sem filtragem) foi feito o diagrama espectral de Fourier (para o código de Matlab dirija-se ao Apêndice B) para visualização dos principais componentes das frequências destes sinais. Para todos os casos enxergou-se um gráfico como da Fig. 3.1.1.

Figura 3.1.1. Espectro de magnitude de Fourier para o conjunto tensão-corrente da soldagem MIG/MAG medida e manipulada com o robô da MOTOMAN.



Fonte: Próprio autor.

No gráfico da Fig. 3.1.1, pode-se observar com clareza que a maior densidade de potência do conjunto tensão-corrente localiza-se no nível de corrente contínua que é entorno de 4,4 vezes maior que os sinais espúrios enxergados entre 1 – 150 Hz.

Pelo que o maior interesse dos sinais de soldagem medidas deve-se focalizar no seu nível de corrente contínua com uma filtragem de quarta ordem com frequência de corte localizada em 25 Hz, o que garantirá a filtragem necessária do sinal de soldagem para monitoramento.

Como exemplo para o caso da Fig. 3.1.1, a densidade da potência do sinal na frequência de 45 Hz é de 0,5 (normalizada) para um sistema de filtragem de Chebyshev de quarta ordem com frequência de corte em 25 Hz, a correspondente atenuação será então:

$$Pot(dB) = \left(-\frac{40dB}{dec} \right) \times \left(\frac{45Hz}{225Hz} \right) - 3dB = -11dB \quad (3.1.1)$$

$$Pot(dB) = 10 \log(Pot(W)) \quad (3.1.2)$$

Com esse resultado pode-se calcular o nível de atenuação em termos lineares

$$Potência\ do\ sinal = Pot(W) \times 10^{-1.1} = 0,08 \times Pot(W) \quad (3.1.3)$$

Com o resultado então a potência do sinal em 45 Hz será

$$Potência\ do\ sinal\ (45Hz) = 0,08 \times 0,5 = 0,04 \quad (3.1.4)$$

As potências dos sinais acima desta frequência terão uma atenuação ainda maior dado que o filtro de quarta ordem atenua a razão de -40dB por década (SEDRA, 2006). No caso do resultado da Eq. (3.1.4) a potência do sinal medida em 45Hz será 55 vezes menor que a potência desse mesmo sinal de interesse em 0Hz.

3.2 Materiais

Baseado no esquemático proposto na Fig. 1.2.1, e as ideias propostas no apartado 3.1 fizeram-se uma lista de componentes e equipamentos divididos em duas partes, os componentes com os quais se contavam e os componentes a adquirir.

Tabela 3.2.1 Componentes e equipamentos do Laboratório de Pesquisa em Engenharia de Soldagem (LAPES).

Qtd.	Modelo	Descrição
1	SDS1062CM	Osciloscópio Digital
1	POWER WAVE 455M/STT	Fonte de Soldagem MIG/MAG
1	POWER FEED 10M	Alimentador de arame
1	PMC400-2K	Tocha de soldagem MIG/MAG
1	Chapa de alumínio 6mm	800 X 400 mm
1	Tubo de aço	60 X 100 mm
1	Torno	-

Fonte: Próprio autor.

Entre os componentes adquiridos, encontram-se microcontroladores, pensando em um sistema autônomo que possa ser ligado sem a necessidade de transportar um computador para o seu funcionamento entre outros componentes necessários para a implantação do sistema mecatrônico.

Tabela 3.2.2. Componentes utilizados para o projeto.

Qtd.	Modelo	Descrição
1	PIC16F877A	μC para o controle da velocidade
1	PIC18F4550	μC para o controle do posicionamento
1	PIC10F322	μC para o controle correção da trajetória
1	MD-1301	Multímetro Digital
1	PIC kit 3	Programador μC
1	PK-150	Programador μC
2	US-100	Sensor por ultrassom
2	SG90	Servomotor
1	Moto-reductor	Motor CC
2	Polias-correia	Acoplamento motor CC
1	FPC-2410	Fonte de CC 24V – 10A
20	Terminais	Terminal eletrônica
50	Resistores	Resistências eletrônicas
20	Capacitores	Capacitâncias eletrônicas
4	L6203	Driver motor de passo
1	2N60	MOSFET
1	Encoder	Sensor de velocidade
150	M3-M4-M6	Parafusos e porcas de Aço temperado
2	Macho/Cossinete M4 – M6	Ferramenta para fazer rosca
2	LM7805	Regulador de tensão
50	XH2.54	Terminais eletrônicas
1	MAX7219	Controlador Display 7 segmentos
4	Display 7 segmentos	Displays
8	Botões	Para interface com o usuário
20	Separadores - Suportes	Separador M3 X 50 mm
2	Acoplamentos alumínio	Acoplamento motor de passo 6 X 8 mm
1	UV-Light	Lâmpada de luz ultravioleta
8	Placas de cobre	Placas virgens para circuito impresso
1	Furadeira manual	Furadeira 12V-1A
10	Brocas PCB	Brocas para circuito impresso
10	Películas PCB UV	Películas foto sensitivas a UV para fabricação de PCB
1	Barra roscada	Barra roscada para a mesa linear
1	Eixo trapezoidal - castanha	Eixo 8 mm com a castanha para o sistema de deslocamento angular
4	SS604-ZZ	Rolamento 4mm
2	SS608	Rolamento 8mm
4	Mancais 8 mm	Mancal para as mesas lineares
4	Suportes 10 mm	Suportes para as guias lineares
2	Guias lineares 10 mm	Guias lineares para a mesa
1	GP2Y0A21YK	Sensor por Infravermelho
1	Ferro de solda	Ferro para brasagem eletrônica

Fonte: Próprio autor.

Cabe a ressalva à não escolha de sistemas baseados no software Arduino, devido a que é um sistema com licenciamento aberto (e de uso livre), o que pode gerar facilmente a clonagem do sistema ou mecanismo a se projetar.

Figura 3.2.1. Ferramentas do LAPES e algumas adquiridas para o projeto.



Fonte: Próprio autor.

3.3 Metodologia

A metodologia a seguir para o desenvolvimento do sistema mecatrônico, primeiramente terá as etapas de projeto dos sistemas mecânicos e eletrônicos, seguidamente a etapa de montagem do sistema e por fim testes do sistema mecatrônico com a tocha de soldagem MIG/MAG em peças metálicas de aço em configuração T. Nos capítulos seguintes apresentam-se os procedimentos para o projeto e montagem do sistema.

As listas de componentes apresentadas nas Tab. (3.2.1) e Tab. (3.2.2) foram feitas analisando as necessidades já ditas da praticidade e simplicidade do sistema a criar.

Dado que cada processo de soldagem apresenta parâmetros de projeto diferentes dependendo, por exemplo, do material a soldar, espessura das chapas, posição da soldagem, deixou-se como única opção ajustável pelo usuário a velocidade de soldagem. Tendo que o mecanismo será programado para posicionar automaticamente a tocha, para manter a

velocidade de soldagem estável, e para corrigir (no caso ocorram) distorções ou desalinhamentos das peças na direção de geração do cordão de solda.

Tudo isso com o intuito de minimizar a intervenção do técnico no processo e diminuir os períodos de capacitação no uso do equipamento.

Capítulo 4. Projeto do Sistema Mecânico

4.1 Dimensionamento do sistema mecânico

O sistema de forma geral terá um tamanho e forma estabelecida principalmente pelas ferramentas e equipamentos com as quais se contam para a construção das peças que formarão o sistema mecatrônico (robô de soldagem) para o posicionamento da tocha de soldagem.

Um dos principais componentes do sistema é a fonte de alimentação, que fornecerá de energia aos subsistemas eletrônicos e eletromecânicos, que segundo os cálculos feitos de potência necessária para o sistema total, terá as dimensões como alistadas na Tab. (4.1.1).

Outros componentes que estarão dentro do sistema serão os servomotores, o motor de corrente contínua que movimentará o robô de soldagem e os circuitos eletrônicos. Todos esses componentes mencionados serão necessários para o correto funcionamento do sistema mecatrônico, por isto é preciso que o dimensionamento seja de forma integral. Todas as dimensões dos principais componentes se alistam na Tab. (4.1.1).

Tabela 4.1.1. Dimensões dos componentes principais do sistema mecatrônico.

Quantidade	Componente	Comprimento (mm)	Largura (mm)	Altura (mm)
1	Fonte de potência	180	102	36
1	Circuito driver motor de CC	65	40	25
1	Circuito CNC	100	50	20
1	Circuito principal	100	65	20
1	Circuito dos servomotores	50	40	20
2	Servomotores	22	12	29
1	Motor de CC	80	38	38
1	Circuito da interface do usuário	88	55	20
1	Circuito para visualização	110	70	25
1	Tocha de soldagem	240	38	38
2	Sensores por ultrassom	45	25	17

Fonte: Próprio autor.

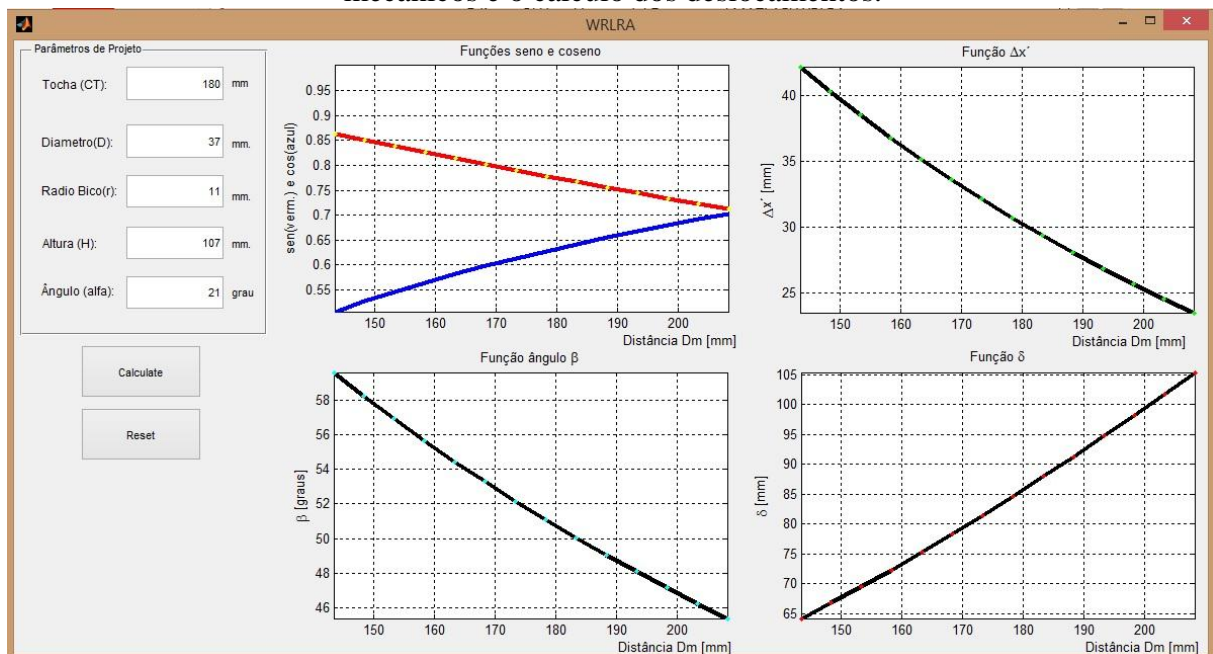
O sistema contém todos os componentes mencionados e a forma de localização dentro do carrinho será estratégica priorizando a funcionalidade do mecanismo, além disso, tentando

prever a estabilidade do sistema no andamento mantendo o *centro de massa o mais próximo ao solo e o mais centrado possível*.

Com ajuda das ferramentas de software Solidworks® e Matlab™ foi feito o layout do mecanismo a ser implantado considerando as dimensões e demais parâmetros já ditos como a estabilidade, robustez, etc., porém priorizando a funcionalidade do sistema integral com ajuda do computador segundo as metodologias empregadas por Erdman (1998 p. 11-18; 96-101) e o Sandler (1999 p. 45).

Para ter uma visão integral dos deslocamentos mecânicos, alturas dos componentes e tamanhos necessários das mesas lineares foram obtidas e montadas no Matlab™ (para o código dirija-se ao apêndice B) as equações de projeto mecânico (baseado nas geometrias do mecanismo obtido no Solidworks®) e foi desenvolvida uma interface de usuário para automatizar o processo de re-projetar, avaliando a estrutura baseado nas dimensões da tocha de soldagem, a altura do carrinho, a altura da mesa linear e o ângulo de soldagem e do mecanismo angular.

Figura 4.1.1. Interface do usuário para projeto o mecânico das dimensões dos sistemas mecânicos e o cálculo dos deslocamentos.



Fonte: Próprio autor.

A interface do usuário do programa para o projeto mecânico é apresentada na Fig. 4.1.1 e permite inserir os dados físicos da altura do robô, diâmetro da tocha de soldagem, raio

do bocal da tocha, o ângulo inicial do mecanismo angular e o comprimento da tocha de soldagem, todos esses dados fornecidos pelo projetista mecânico. Então, por exemplo, tendo uma tocha com comprimento (CT) de 180 mm, um diâmetro da tocha (D) de 37 mm, um raio do bico de contato (r) de 11 mm e uma altura do centro instantâneo do mecanismo angular (H) montado encima do carrinho de 107 mm, o programa automaticamente gera as seguintes informações.

- **Função $\Delta x'$** : Deslocamento necessário do cursor do mecanismo angular para atingir o ângulo de soldagem baseado na distância do carrinho até a peça a soldar. Para o exemplo o deslocamento máximo do cursor foi de 45 mm, pelo que essa informação podia ser inserida no desenho no Solidworks®.

- **Função δ** : Deslocamento linear da mesa que suporta a tocha de soldagem para atingir a peça no posicionamento inicial do processo de soldagem, baseado na distância do carrinho até a peça, na altura do mecanismo e do ângulo de soldagem. Para o exemplo o deslocamento máximo da tocha devia ser 105 mm, então o mecanismo linear deveria ter no mínimo essa quantidade de avanço para que a tocha pudesse atingir a peça.

- **Função β** : Ângulo de soldagem da tocha, para uma distância dada entre o carrinho e a peça a soldar, e a altura do mecanismo.

O sistema mecânico completo será dividido em quatro sub-mecanismos para facilitar o processo de fabricação e controle. O projeto de cada sub-mecanismo é detalhado a seguir.

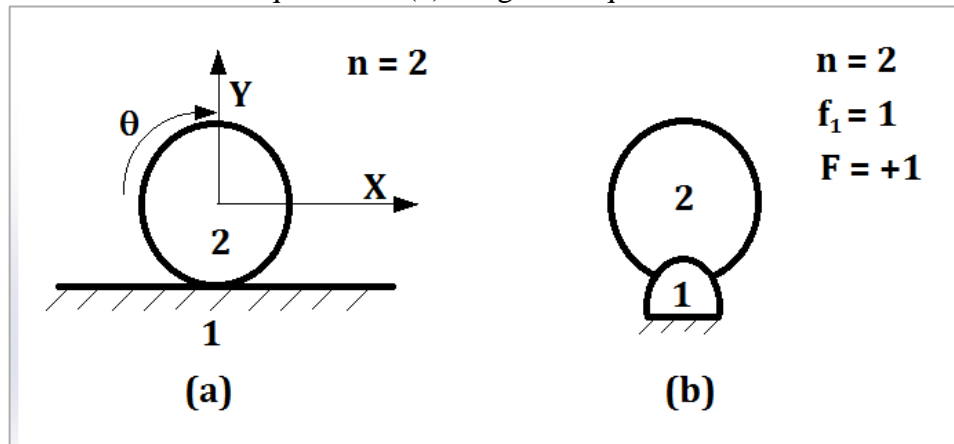
4.2 Mecanismo de velocidade constante

O Diagrama do par cinemático de plano-junta do mecanismo de velocidade constante se apresenta na Fig. 4.2.1.

A equação de *Gruebler* que define os graus de liberdade do mecanismo (ERDMAN, 1998 p. 24) apresenta-se a seguir:

$$\text{graus liberdade} = F = 3(n - 1) - 2f_1 \quad (4.2.1)$$

Figura 4.2.1. Par cinemático plano-junta dos elos do sistema de velocidade constante, (a) Esquemático (b) Diagrama equivalente.



Fonte: Próprio autor.

Para o caso do sistema da Fig. 4.2.1, o mecanismo apresenta um único grau de liberdade. Um aspecto importante no projeto do mecanismo de velocidade é a velocidade resultante do sistema que será estabelecida dentro dos limites expostos na Tab. (4.2.1).

Tabela 4.2.1. Velocidades utilizadas para o projeto mecânico.

	[cm/min]	[Rpm]
Velocidade máxima	110	5,57
Velocidade mínima	15	0,76

Fonte: Próprio autor.

A geometria da roda a desenhar para o mecanismo vai considerar a máxima velocidade devido à redução da velocidade pela modulação da largura do pulso (PWM). A equação da velocidade baseado na geometria da roda é mostrada a seguir.

$$V_{el} = \frac{2\pi r}{tempo} \quad (4.2.2)$$

Isolando o raio da roda da Eq. (4.2.2) obtém-se

$$r_{roda} = \frac{V_{el} \times tempo}{2\pi} = \frac{110 \left[\frac{cm}{min} \right] \times 1 [min]}{2\pi} = 17,51 [cm/rad] \quad (4.2.3)$$

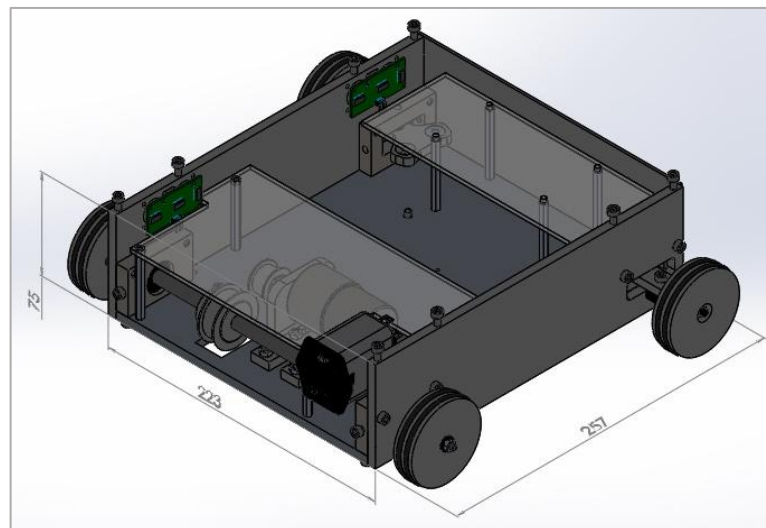
Convertendo para centímetros se tem o valor do raio mínimo da roda para o projeto mecânico.

$$r_{roda} = \frac{17,51 [cm] [rad]}{[rad] \frac{2\pi}{}} = 2,79 cm \quad (4.2.4)$$

Sabendo que pode ser definido um raio maior dado que se tem o controle de velocidade por PWM, se aumentará ligeiramente o raio da roda para a escolha de um motor com maior torque, dito isto o raio escolhido será de 3 cm. Importante ressaltar ao fato que é *necessário definir a geometria do mecanismo ou será é impossível calcular os esforços mecânicos.*

Com ajuda do software Solidworks® e os dados das Tab. (4.1.1) e Tab. (4.2.1), os resultados obtidos na análise feito na Fig. 4.2.1, e da Eq. (4.2.4), traçou-se o desenho com as dimensões mínimas para o sistema de velocidade constante que se apresenta na Fig. 4.2.2.

Figura 4.2.2. Desenho do sistema mecânico de velocidade constante com o Solidworks®, as dimensões estão em milímetros.

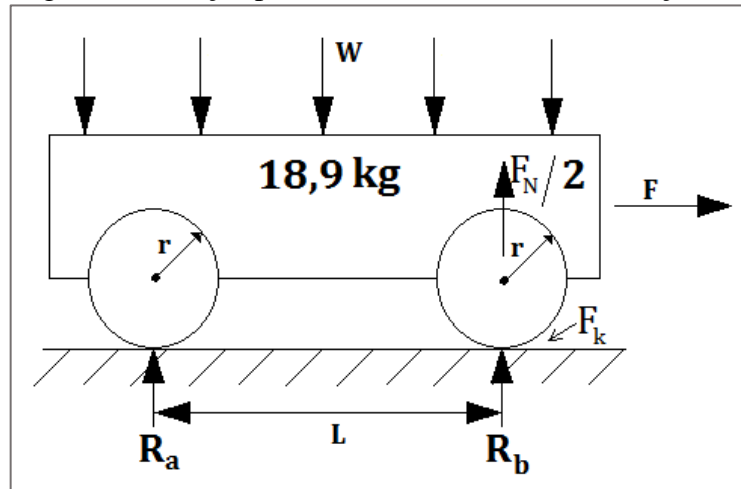


Fonte: Próprio autor.

A Fig. 4.2.2 mostra o desenho do mecanismo de velocidade de soldagem baseado nas dimensões dos componentes apresentados na Tab. (4.1.1) e o raio das rodas da Eq. (4.2.4). Para conseguir fazer esse desenho mecânico foi necessário considerar também os espaçamentos físicos necessários entre os componentes internos, tais como os sensores, placas de circuito eletrônico, a fonte de energia e o tamanho dos servomotores. O diâmetro total das rodas foi de 6 cm incluindo um arranjo de anéis de borracha para dar tração ao sistema e evitar o escorregamento pelo contato metal-metal.

O diagrama utilizado para o cálculo dos esforços e dimensionamento do motor e a relação das polia-correia apresenta-se na Fig. 4.2.3.

Figura 4.2.3. Diagrama de forças para o cálculo do motor e a relação das polias-correia.



Fonte: Próprio autor.

Para o cálculo dos esforços será necessário enumerar alguns parâmetros intrínsecos e extrínsecos do mecanismo, a Tab. (4.2.1) resume essas constantes. Para o cálculo da força normal precisa-se o peso do sistema o qual vai considerar que o volume inteiro do mecanismo da Fig. 4.2.2 estará totalmente preenchido pelos diferentes componentes e peças alistadas na Tab. (4.1.1), no entanto os componentes eletrônicos como circuitos e motores são mais leves do que o alumínio, pelo que o peso total do mecanismo incluindo todos os componentes pode-se aproximar a seguir.

$$m = V_{ol} \times D_{al} + 4\pi r^2 \times D_{aço} \times L_r + m_t \quad (4.2.5)$$

Tabela 4.2.1. Constantes definidas para o cálculo dos esforços da Fig. 4.2.3

Parâmetro	Símbolo	Valor
Atrito borracha-metal estático	μ_s	0,6
Distância entre eixos	L	0,15 m
Atrito borracha-metal dinâmico	μ_d	0,15
Densidade do alumínio	D_{al}	2700 kg/m ³
Densidade do aço	$D_{aço}$	7700 kg/m ³
Raio	r	0,03 m
Largura da roda	L_r	0,015 m
Volume ¹ do carro	V_{ol}	0,0043 m ³
Massa da tocha + sistema angular	m_t	6 kg
Aceleração gravitacional	g	9,8 m/s ²
Relação das polias XL	R	2

Fonte: (SHIGLEY, 2005, p. 902) e próprio autor.

¹ Calculado a partir das dimensões da Fig. 4.2.2.

Substituindo os valores correspondentes na Eq. (4.2.5) se obtém o peso total do sistema

$$m = 11,6kg + 1,3kg + 6kg = 18,9kg$$

A força normal do sistema é dada pela Eq. (4.1.6) (SHIGLEY, 2005 p. 46)

$$F_N = m \times g \quad (4.2.6)$$

Substituindo os valores correspondentes na equação anterior obtém-se

$$F_N = 18,9 kg \times 9,8m/s^2 = 185,2 N$$

Do diagrama de corpo livre para a roda b, a força neste ponto é:

$$R_b = \frac{F_N}{2} = 92,6 N \quad (4.2.7)$$

A força necessária a vencer no momento da partida é dada pela relação da força da Eq. (4.2.7) e o coeficiente de atrito estático entre o aço da peça horizontal e a borracha dos anéis das rodas, isto é:

$$F_s = R_b \times \mu_s \quad (4.2.8)$$

Substituindo os valores correspondentes na Eq. (4.2.8) obtém-se a força necessária em cada roda para movimentar o sistema na partida.

$$F_s = 92,6N \times 0,6 = 55,56N$$

O momento de inércia no eixo das rodas é dado pela Eq. (4.2.9) (SHIGLEY, 2005 p. 119)

$$M = F \times r \quad (4.2.9)$$

Substituindo os valores correspondentes na Eq. (4.2.9) obtém-se o momento de inércia a se vencer na partida do motor, tendo $F = F_s$.

$$M_s = 55,56N \times 0,03m = 1,67 N \cdot m$$

Pelas limitações de espaço dentro do mecanismo da Fig. 4.2.2, escolheu-se colocar no eixo uma polia com as dimensões apresentadas na Fig. 4.2.4 (a).

A Fig. 4.2.4 (b) mostra o espaço disponível dentro do mecanismo para uma polia com as dimensões escolhidas, na montagem foi necessário recortar a base do mecanismo para acoplar a polia no eixo, tendo que esta é a polia maior que se pode instalar no eixo, a relação para esse padrão de polias é de 2:1.

Desta forma o momento de inércia no eixo do motor na partida, considerando a relação mecânica das polias é dado pela relação R (como foi apresentado na Tab. (4.2.1)) para um conjunto correia-polia sincrônica (SHIGLEY, 2005 p. 841).

$$M_{ms} = \frac{M_s}{R} = \frac{1,67 \text{ N} \cdot \text{m}}{2} = 0,835 \text{ N} \cdot \text{m} \quad (4.2.10)$$

A força no eixo das rodas após a partida (em movimento) é dada pela Eq. (4.2.11).

$$F_d = R_b \times \mu_d \quad (4.2.11)$$

Substituindo os valores correspondentes na Eq. (4.2.11), pode-se calcular a força dinâmica do sistema como é mostrado a seguir:

$$F_d = 92,6 \text{ N} \times 0,15 = 13,9 \text{ N}$$

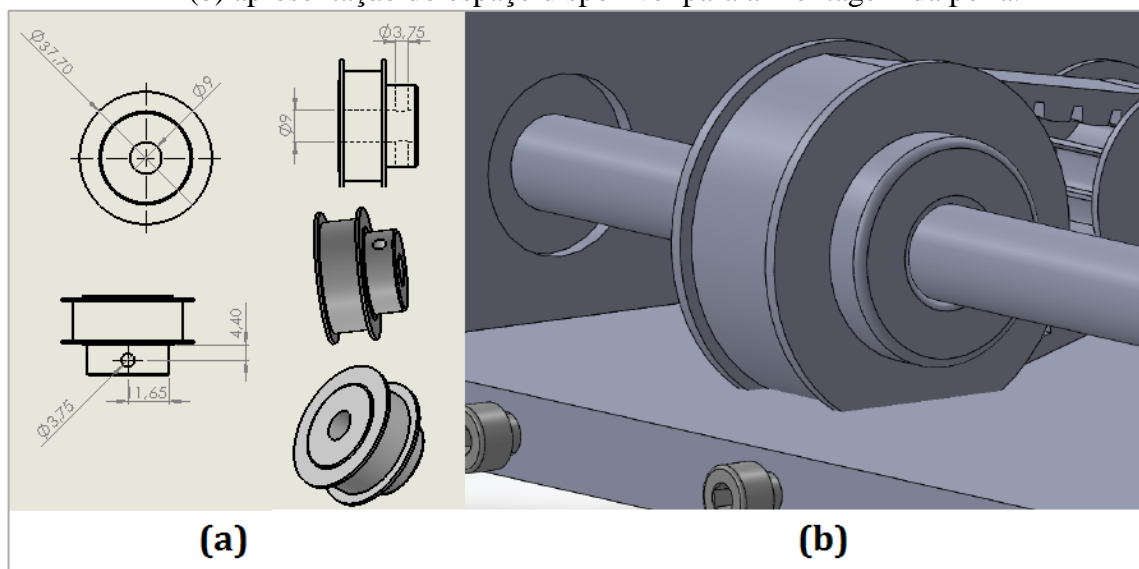
O momento de inércia no eixo das rodas após a partida é calculado com ajuda da Eq. (4.2.9), tendo que $F = F_d$.

$$M_d = 13,9 \text{ N} \times 0,03 \text{ m} = 0,417 \text{ N} \cdot \text{m} \quad (4.2.12)$$

De novo o momento de inércia no eixo do motor após a partida, considerando a relação mecânica das polias que é dada pela relação R

$$M_{md} = \frac{M_d}{R} = \frac{0,417 \text{ N} \cdot \text{m}}{2} \cong 0,209 \text{ N} \cdot \text{m} \quad (4.2.13)$$

Figura 4.2.4. Polia escolhida para montagem no eixo das rodas. (a) Dimensões da polia em mm (b) apresentação do espaço disponível para a montagem da polia.



Fonte: Próprio autor.

Finalmente as características do motor a montar no sistema se resumem na Tab. (4.2.2).

Tabela 4.2.2. Parâmetros do motor de corrente contínua.

Parâmetro	Símbolo	Valor calculado	Valor medido no motor
Torque partida	T_{ms}	$0,835 N \cdot m$	$2,5 N \cdot m$
Torque nominal	T_{md}	$0,209 N \cdot m$	$0,833 N \cdot m$
Velocidade de rotação (2:1)	V_{el}	$11,14 rpm$	$13 rpm$

Fonte: Próprio autor.

4.3 Equações para a análise da resistência à fadiga mecânica

O critério DE-Gerber (SHIGLEY, 2005 p. 877) para fadiga mecânica é apresentado a seguir:

$$\frac{1}{n_{seg}} = \frac{16K_f M_a}{\pi d^3 S_e} \left\{ 1 + \sqrt{1 + 3 \left(\frac{K_{fs} T_m S_e}{K_f M_a S_{ut}} \right)^2} \right\} \quad (4.3.1)$$

As constantes de concentração de tensão são dadas a seguir

$$K_f = 1 + q(K_t - 1) \quad (4.3.2)$$

$$K_{fs} = 1 + q_{shear}(K_{ts} - 1) \quad (4.3.3)$$

Para o aço de liga 1045 segundo a Tab. (A-20) do Shigley (2006, p. 1023), o S_{ut} é 220 *kpsi* ou 1520 *MPa*.

O limite de resistência da peça é dado da forma:

$$S_e = k_a k_b k_c k_d k_e k_f S'_e \quad (4.3.4)$$

Do Shigley (2006, p.274), se tem que para aços com o $S_{ut} > 212 \text{ kpsi}$

$$S'_e = 100 \text{ kpsi ou } 700 \text{ MPa} \quad (4.3.5)$$

Para a Eq. (4.3.4) as constantes modificadoras da tensão se calculam a seguir:

$$k_a = a S_{ut}^b \quad (4.3.6)$$

$$k_b = 1,24 d^{-0,107} \quad (4.3.7)$$

$$k_c = k_d = k_f = 1 \quad (4.3.8)$$

Para uma confiabilidade do 99,99% (SHIGLEY, 2005, p. 327)

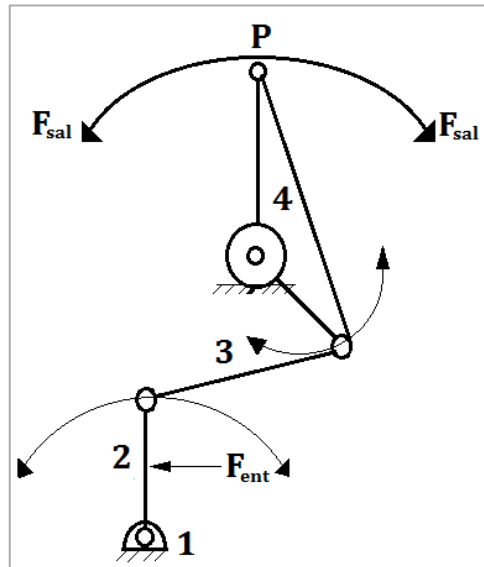
$$k_e = 0,72 \quad (4.3.9)$$

4.4 Mecanismo da direção

O mecanismo da direção é um conjunto de peças simples, que deslocará através de um servomotor uma alavanca que mudará o ângulo das rodas dianteiras para ajustar a trajetória do carrinho, e tentar manter uma linha reta paralela à peça vertical a ser soldada.

Para o desenho deste mecanismo foi utilizada a ferramenta de software Solidworks®, o mecanismo projetado devia ser simples devido às limitações do equipamento para a fabricação da peça.

Figura 4.4.1. Diagrama movimento de 4 elos e 4 barras com um grau de liberdade.



Fonte: Modificado de Erdman (1998, p. 8).

Partindo da Eq. (4.2.1) podem-se encontrar os graus de liberdade do mecanismo

$$\text{graus liberdade} = F = 3(4 - 1) - 2(4) = +1$$

4.4.1 Cálculos dos esforços no mecanismo da direção.

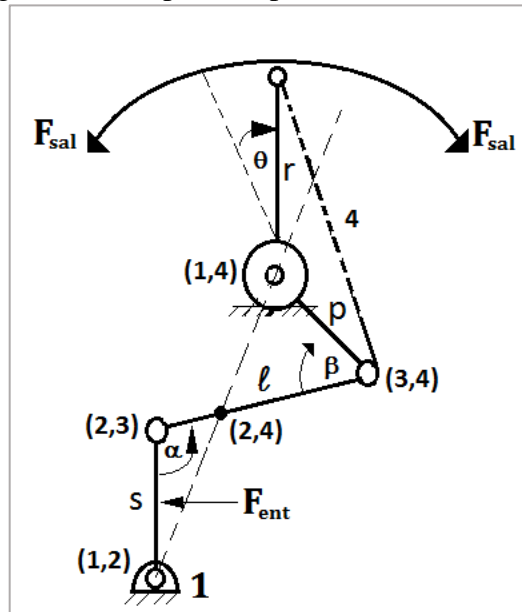
O cálculo do esforço vai considerar que o mecanismo não apresenta nenhuma força de flexão nos elos do sistema ou carregamento axial dos rolamentos colocados nos elos, vistos na Fig. 4.4.1.

A magnitude do esforço de saída pode ser calculada integrando a análise precedente feita com ajuda da Eq. (4.2.11) o resultado dessa força vai ser:

$$\overline{F_{sal}} = \frac{F_d}{2} = 7 [N] \quad (4.4.1)$$

Porém é necessário incluir na análise a componente radial do mecanismo e para isto será utilizado o diagrama de corpo livre da Fig. 4.4.2.

Figura 4.4.2. Diagrama de corpo livre para o cálculo dos esforços e do motor.



Fonte: Próprio autor.

$$F_{sal} = \frac{F_d}{2} e^{i\theta} = 7 \cdot e^{i\theta} [N] \quad (4.4.2)$$

No entanto a Eq. (4.4.2) pode-se simplificar considerando que somente a parcela real é considerada no cálculo da força resultante, então a Eq. (4.4.2) pode ser reescrita com ajuda da identidade de Euler (ERDMAN, 1998, p. 140), então, a nova fórmula para o esforço de saída é:

$$F_{sal} = 7 \cdot \cos(\theta) [N] \quad (4.4.3)$$

Para o caso de mecanismos conformados por elos e barras deve-se cumprir que:

$$P_{sal} = F_{sal} V_{sal} = F_{ent} V_{ent} = P_{ent} \quad (4.4.4)$$

E para mecanismos deste tipo a vantagem mecânica é calculada como (ERDMAN, 1998, p. 166).

$$V.M. = \frac{F_{sal}}{F_{ent}} = \left(\frac{\omega_{ent}}{\omega_{sal}} \right) \left(\frac{s}{r} \right) \quad (4.4.5)$$

Desenvolvendo a Eq. (4.4.5) em termos das velocidades do mecanismo

$$F_{ent} = F_{sal} \left(\frac{r}{s} \right) \left(\frac{[1,4]-[2,4]}{[1,2]-[2,4]} \right) \quad (4.4.6)$$

Substituindo pelos comprimentos correspondentes para encontrar os valores necessários para a fabricação do mecanismo, suponha-se $\alpha = 90^\circ$ e $\beta = 60^\circ$, a equação resultante para o cálculo da força de entrada é dada a seguir:

$$F_{ent} = F_{sal} \left(\frac{r}{s} \right) \left(\frac{2p}{2s} \right) = F_{sal} \left(\frac{r \cdot p}{s^2} \right) \quad (4.4.7)$$

Na Fig. 4.4.2. r , p e s foram estabelecidos nos valores de 35, 20 e 35 mm respectivamente:

$$F_{ent} = F_{sal} \left(\frac{(35)(20)}{35^2} \right)$$

$$F_{ent} = \frac{4}{7} F_{sal} = 4 \cdot \cos(\theta) [N] \quad (4.4.8)$$

O momento de inércia no eixo do motor agora pode ser calculado a seguir baseado na Eq. (4.4.9), considerando o valor de s em 3 cm

$$M_{servo} = F_{ent} \cdot s = 4 \cdot s \cdot \cos(\theta) [N \cdot cm] \quad (4.4.9)$$

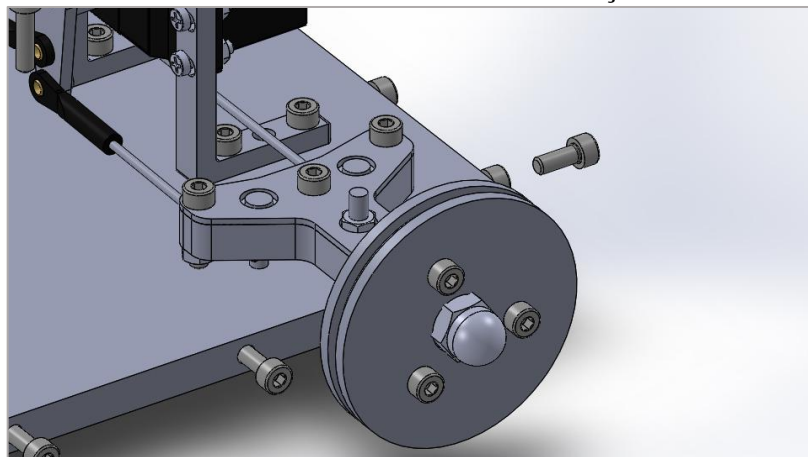
Definindo o valor máximo de $\theta = 30^\circ$, então

$$T_{servo} = 4(3) \cdot \cos(30^\circ) = 10,4 [N \cdot cm]$$

$$T_{servo} = 1,06 [kg \cdot cm]$$

Os servomotores escolhidos apresentam um torque nominal de 1,5 kg-cm. O desenho criado no software Solidworks® para o mecanismo da direção apresenta-se na Fig. 4.4.3.

Figura 4.4.3. Desenho em 3D do mecanismo da direção com Solidworks®.

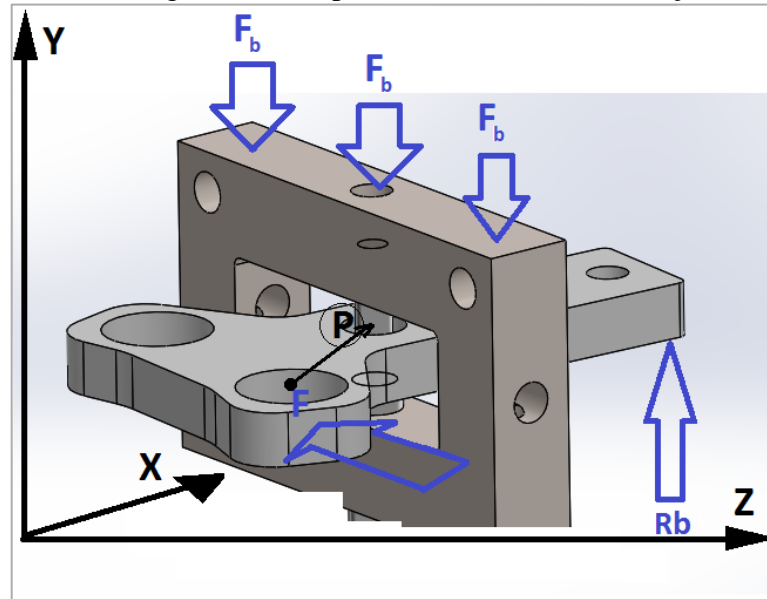


Fonte: Próprio autor.

4.4.1 Cálculos das dimensões do mecanismo da direção.

O sistema da direção projetado apresenta a configuração dos esforços mecânicos como se enxerga na Fig. 4.4.4.

Figura 4.4.4. Diagrama de corpo livre do sistema da direção em 3D.



Fonte: Próprio autor.

Para o sistema da Fig. 4.4.4, tem-se que:

$$\sum F_y = F_b = R_b \quad (4.4.10)$$

O momento de torção no ponto **P** é calculado como:

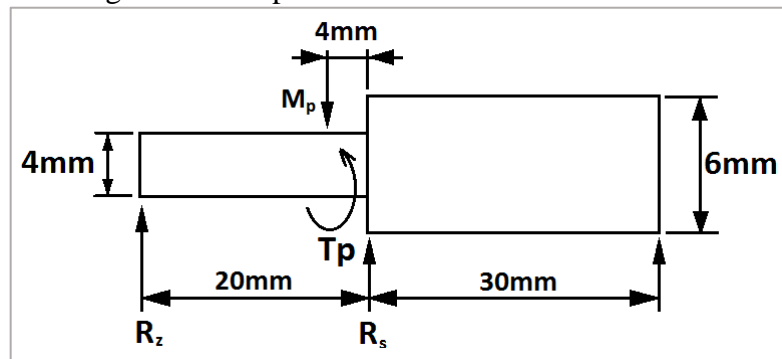
$$T_p = F \cdot p \quad (4.4.11)$$

O momento de flexão no ponto **P** é calculado como:

$$M_p = F_b \cdot r \quad (4.4.12)$$

Neste mecanismo têm-se duas peças no ponto **P** que estão expostas à concentração do esforço e poderão apresentar alguma falha. A primeira peça é o eixo vertical que suporta o mecanismo completo, para essa peça avaliam-se da Eq. (4.3.1) até a Eq. (4.3.9). Tendo que $K_t = 1,8$ (SHIGLEY, 2006 p. 1008), para o tipo de aço SAE 1045, q , extraiu-se da figuras 6-20 do SHIGLEY (2006, p. 288-289), e o seu valor é de 0,9. O diagrama de corpo livre para o eixo vertical apresenta-se na Fig. 4.4.5.

Figura 4.4.5. Diagrama de corpo livre do eixo vertical do mecanismo da direção.



Fonte: Próprio autor.

O momento flexor no ponto crítico de maior concentração de tensão é denotado por M_p na Fig. 4.4.5, que pode ser aproximado com ajuda da Eq. (4.4.12). O momento de torção que é denotado por T_p é zero, dado que nesse ponto colocou-se um rolamento para diminuir os efeitos do atrito no giro do mecanismo. Com os valores expostos tem-se:

$$K_f = 1 + 0,9 \cdot (1,8 - 1) = 1,72$$

$$T_p \approx 0$$

A força F_b da Eq. (4.4.12) é equivalente à quarta parte da força normal da Eq. (4.2.6)

$$M_p = \frac{F_N}{4} \cdot r = (46N)(30mm) = 1380N \cdot mm$$

$$S'_e = 700 MPa$$

$$k_a = 0,8$$

$$k_b = k_c = k_d = k_f = 1$$

$$k_e = 0,72$$

$$S_e = (0,8)(1)(1)(1)(0,72)(1)(700) = 403 MPa$$

Isto é, substituindo todos os resultados precedentes na Eq. (4.3.1)

$$\frac{1}{n_{seg}} = \frac{32(1,72)(1380)}{\pi(6)^3(403)} = 0,27$$

Para um coeficiente de segurança $n_{seg} = 3,6$.

A segunda peça é o rolamento do padrão 604zz, que apresenta na operação carregamento axial. A faixa estática de carregamento radial C_0 é 280N para o rolamento

604zz. Para o rolamento do tipo radial o carregamento equivalente é dado pela equação a seguir (SHIGLEY, 2006 p.559).

$$F_e = X_i V F_r + Y_i F_a \quad (4.4.13)$$

Tendo que a relação de carregamento radial e axial apresenta o valor calculado com a equação a seguir:

$$\frac{F_a}{C_0} = \frac{F_b}{280N} = \frac{F_N/4}{280N} = \frac{46N}{280N} = 0,16 \quad (4.4.14)$$

Para os cálculos da Eq. (4.4.14) a força radial do rolamento é equivalente a:

$$F_r = \frac{T_{servo}}{p} = \frac{150kg.mm}{20mm} = 7,5kgf = 0,765N \quad (4.4.15)$$

Para rolamentos com rotação do anel exterior o fator V toma o valor de 1,2. (SHIGLEY 2006, p. 560), e utilizando os resultados das Eq. (4.4.15) e Eq. (4.4.14).

$$\frac{F_a}{V F_r} = \frac{46N}{(1,2)(0,765)} > e$$

Os valores para (X_2, Y_2) , foram extraídos do Shigley (2006, p. 561), dito isto então a Eq. (4.4.13) tomará o valor:

$$F_e = X_2 V F_r + Y_2 F_a = (0,56)(1,2)(0,765) + (1,31)(46) \approx 61N$$

A pressão equivalente para o rolamento é dada pela Eq. (4.4.16) obtida do Shigley (2006, p. 643).

$$P_r = \frac{F_e}{dL} \quad (4.4.16)$$

Com a variável d sendo o diâmetro interno do rolamento e a variável L o comprimento da bucha, dito isto então a Eq. (4.4.16) tomará o valor:

$$P_r = \frac{61N}{(4)(8)} = 0,0625 MPa$$

A velocidade do rolamento pode ser calculada com a Eq. (4.4.17) (SHIGLEY, 2006 p. 644)

$$V_r = \frac{\pi D n}{12} \quad (4.4.17)$$

Sendo D o diâmetro do rolamento e n a velocidade em RPM, dito isto a Eq. (4.4.17) toma o valor:

$$V_r = \frac{\pi(12)(33)}{12} = 103 \text{ mm/min}$$

O fator PV agora pode ser calculado

$$PV = (0,0625)(103) = 6,43 \text{ MPa} \cdot \text{mm/min}$$

Para o análise de fadiga dos rolamentos NSK (DTS, NSK p. 14)

$$L = \left(\frac{C_0}{F_e}\right)^3 = \left(\frac{280N}{61N}\right)^3 = 96 \times 10^6 \text{ rev} \quad (4.4.18)$$

Todos os valores encontrados para o rolamento encontram-se dentro dos parâmetros de operação normais do rolamento (SHIGLEY, 2006 p.645). Finalmente para o mecanismo da direção que contém o rolamento, e as rodas, os valores das constantes de concentração de tensão K_t é 2 (SHIGLEY, 2006 p. 1006).

$$K_f = 2$$

Se tem que para a liga de alumínio 2024, o S_{ut} é 446 MPa (SHIGLEY, 2006 p. 1020)

$$S'_e = 0,5S_{ut} \quad (4.4.19)$$

$$S'_e = 223 \text{ MPa}$$

As constantes modificadoras da concentração de tensão vistas nas Eq. (4.3.6) à Eq. (4.3.9) calculam-se a seguir

$$k_a = (4,51)(446)^{-0,265} = 0,89$$

$$k_b = 1,24(12)^{-0,107} = 0,95$$

$$k_c = k_d = k_f = 1$$

$$k_e = 0,72$$

Tomando-se os valores das equações precedentes (obtidos acima), pode-se calcular o limite de resistência da peça com a Eq. (4.3.5)

$$S_e = k_a k_b k_c k_d k_e k_f S'_e$$

Com $S'_e = 446 \text{ MPa}$, Isto é

$$S_e = (0,89)(0,95)(1)(1)(0,72)(1)(446) = 271 \text{ MPa}$$

Substituindo os resultados precedentes na equação de fadiga usando o critério representado na Eq. (4.4.20), e com os parâmetros do Shigley (2006 p. 1018)

$$n_{seg} = \frac{S_e}{\sigma_a} = \frac{271}{K_f(M_p c / I_x)} \quad (4.4.20)$$

O segundo momento da área foi encontrado com ajuda da ferramenta Solidworks® e foi 143 mm^4 a massa da peça obtida $m = 16,6 \text{ g}$, e $c = 3,25 \text{ mm}$.

$$I_x = 143 \text{ mm}^4 \quad (4.4.21)$$

Substituindo o resultado da Eq. (4.4.21) na Eq. (4.4.20) e sendo conhecidos os valores do momento de inércia (M_p) e o fator concentrador de tensão (K_f)

$$n_{seg} = \frac{271}{(2)(1380)(3,25/143)} = 4,3$$

Para um fator de segurança n_{seg} de 4,3.

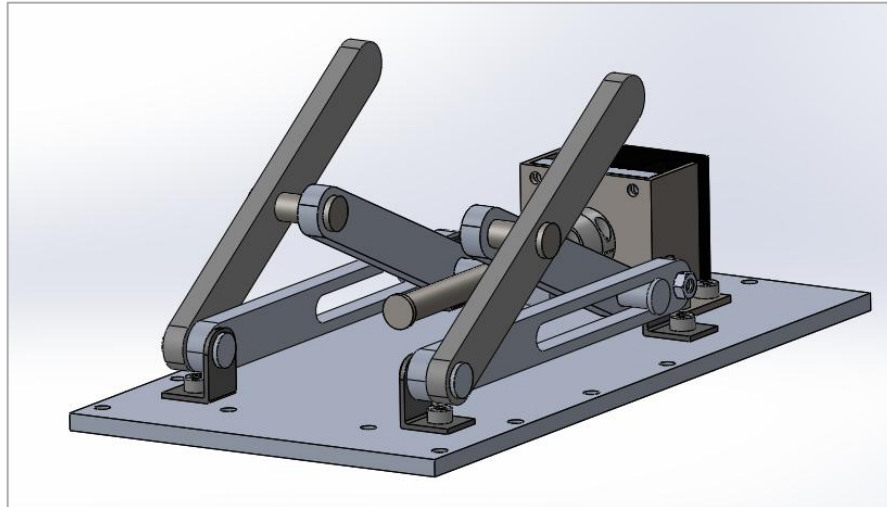
4.5 Dimensionamento mecanismo angular.

Os custos de fabricação dos mecanismos de tipo deslizador-manivela vêm-se reduzidos devido à eliminação da fabricação de superfícies de precisão em levas. A fabricação de levas precisa de um tratamento térmico especial no aço para fornecer a resistência suficiente na superfície. Os deslizadores e acoplamento de deslizador-manivela podem ser fabricados de alumínio fundido a pressão. (ERDMAN, 1998, p. 39).

O sistema angular foi um mecanismo projetado para posicionar a tocha de soldagem no ponto inicial do processo, mudando o ângulo e altura do bico de contato baseado na distância medida entre a peça vertical e o carrinho utilizando o sensor por infravermelho. Além disso, foi pensado em futuras implementações dos movimentos de tecimento angular e na criação do algoritmo SSJ.

O mecanismo completo é apresentado na Fig. 4.5.1, baseou-se num mecanismo do tipo deslizador-manivela.

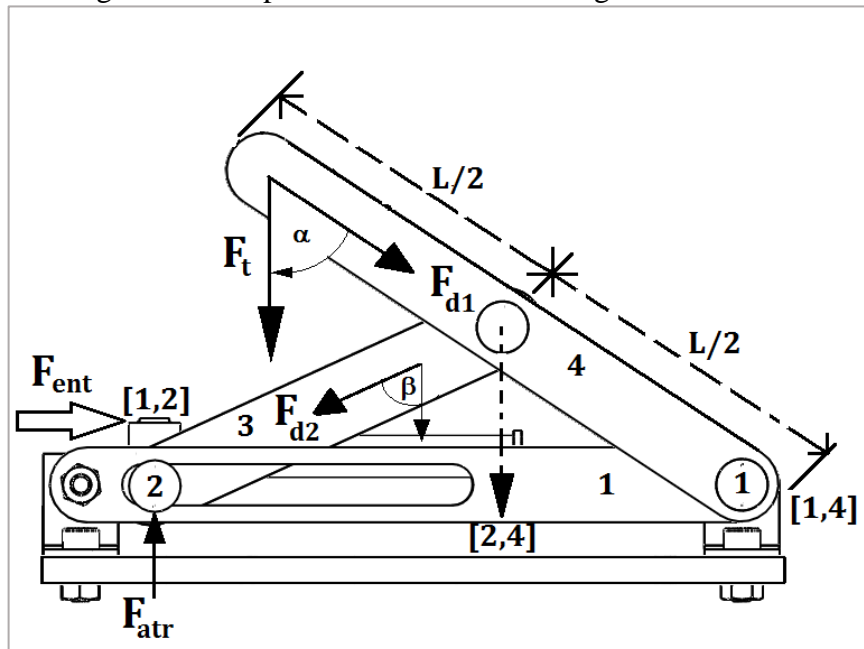
Figura 4.5.1. Mecanismo angular desenhado no Solidworks®.



Fonte: Próprio autor.

Para determinar a potência necessária do motor para este mecanismo fez-se uso do diagrama de corpo livre da Fig. 4.5.2, também incluindo a configuração de barras para obter a relação de velocidades através do método dos centros instantâneos segundo o Erdman (1998).

Figura 4.5.2. Diagrama de corpo livre do mecanismo angular e dos centros instantâneos.



Fonte: Próprio autor.

A força de entrada do mecanismo angular vista no diagrama de corpo livre da Fig. 4.5.2, é somatória dos dispositivos que se alojam em cima do mecanismo. A Tab. (4.5.1) resume esses componentes.

Tabela 4.5.1. Componentes e peças alojadas no mecanismo angular.

Descrição	Peso (kg)
Tocha da soldagem	2,5
Mecanismo completo	2,5
Motores	0,5
Circuitos	0,5
Total	6,0

Fonte: Propio autor.

Com os dados da Tab. (4.5.1), agora se pode calcular a Força F_t , partindo da força distribuída em todo o mecanismo F_{d1} . Os ângulos α e β iniciais do mecanismo terão sempre o valor de 60° , isto porque na prática o menor ângulo de transmissão permissível de um mecanismo deve ser 30° (ERDMAN, 1998, p. 126).

$$F_t = F_{d1} \cdot \cos(\alpha) \quad (4.5.1)$$

$$F_t = 6kg \cdot \frac{9,8m}{s^2} \cdot \cos(60^\circ) \approx 29,5N$$

Para o caso da barra número 3 a força F_{d2} é calculada a seguir:

$$F_{d2} = F_t \cdot \cos(\beta) \quad (4.5.2)$$

$$F_{d2} = 29N \cdot \cos(60^\circ) = 14N$$

Finalmente a somatória de forças no nó da entrada é calculada a seguir:

$$F_{ent} = F_{d2} \cdot \cos(90^\circ - \beta) + F_{atr} \quad (4.5.3)$$

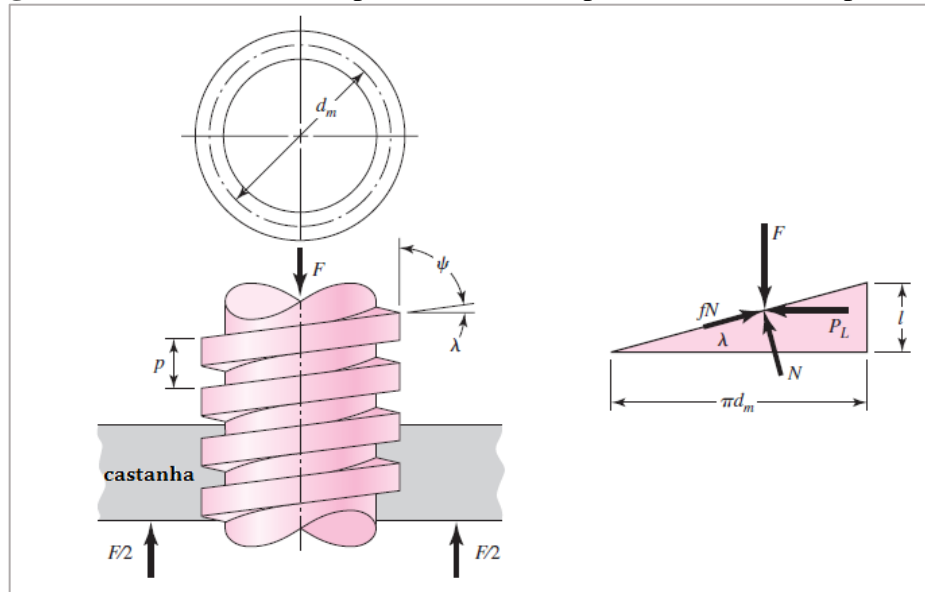
$$F_{ent} = F_{d2} \cdot \cos(90^\circ - \beta) + \mu_{atr} \cdot F_{d2} \cdot \sin(90^\circ - \beta) \quad (4.5.4)$$

Utilizando um coeficiente de atrito μ_{atr} de 0,15 (SHIGLEY, 2006 p. 408), então:

$$F_{ent} = (14N) \cdot \cos(30^\circ) + 0,15 \cdot (14N) \cdot \sin(30^\circ) = 13,17N$$

Esta é a força da entrada que deve vencer o parafuso sem fim movimentado pelo motor de passo. Sabendo que o passo do fuso (parafuso sem fim) empregado é de 5 mm, o diâmetro d_m do fuso é de 8 mm, o ângulo λ de avance de $7,2^\circ$, o ângulo ψ da rosca de $82,8^\circ$ e o coeficiente de atrito f_c é 0,15. Esses parâmetros estão mostrados na Fig. 4.5.3.

Figura 4.5.3. Parâmetros do parafuso sem fim para transmissão da potência.



Fonte: (SHIGLEY, 2006, p. 401).

O par necessário do motor pode-se calcular com a Eq. (4.5.5) (SHIGLEY, 2006, p. 402)

$$T_M = \frac{F_{ent} \cdot d_m}{2} \left(\frac{l + \pi f_c d_m}{\pi d_m - f_c l} \right) + \underbrace{\frac{F_{ent}}{2} Z}_{\text{Inércia barra}} \quad (4.5.5)$$

Onde, Z é a distância entre o elo (2) e o apoio de empurre do motor que é 20 mm.

$$l = \pi d_m \tan(\lambda) \quad (4.5.6)$$

$$l = \pi(4 \text{ mm}) \tan(7,2^\circ) = 1,6 \text{ mm}$$

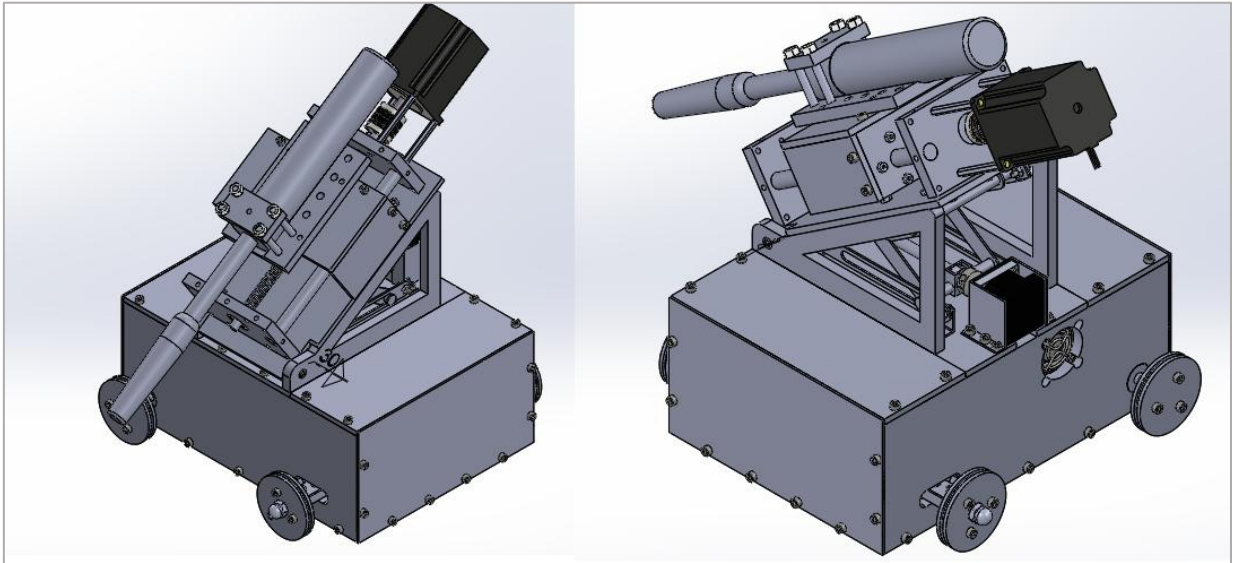
$$T_M = \frac{(13,17 \text{ N}) \cdot (4 \text{ mm})}{2} \left(\frac{1,6 \text{ mm} + \pi(0,15)(4 \text{ mm})}{\pi(4 \text{ mm}) - (0,15)(1,6 \text{ mm})} \right) + (6,6 \text{ N})(20 \text{ mm})$$

$$T_M = 139,44 \text{ N} \cdot \text{mm} = 1,4 \text{ kg} \cdot \text{cm}$$

No caso de motores de passo é recomendável usar motores com um torque pelo menos 2 vezes maior da requerida, para vencer os esforços devido aos desalinhamentos do mecanismo na fabricação, para esta aplicação o valor ideal seria de pelo menos 2,8 kg.cm. **O motor de passo escolhido apresenta um torque maior, sendo o de 5 kg.cm** (caso a tocha de soldagem tenha um peso maior do que o estimado).

Desconsiderou-se os cálculos da mesa linear da parte superior que suportará a tocha de soldagem, considerando a análise precedente, um motor de passo do mesmo torque do sistema angular será suficiente para manejar a tocha de soldagem e posiciona-a no ponto requerido. O mecanismo completo é apresentado na Fig. 4.5.4.

Figura 4.5.4. Sistema mecânico completo em 3D, montado no Solidworks®.



Fonte: Próprio autor.

Capítulo 5. Projeto do Sistema Eletrônico

5.1 Definições preliminares para projetar o sistema de controle

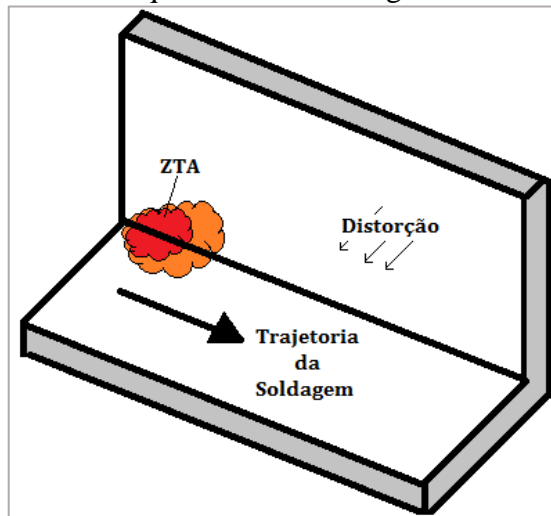
Para a manutenção do arco elétrico a tocha de soldagem deve ser mantida a uma distância e posição constante em relação com a peça a ser soldada, a fim de garantir-se estabilidade e boa qualidade da soldagem final. Na soldagem com equipamento MIG/MAG a estabilidade do processo depende de vários fatores entre estes a transferência do metal de adição, que segundo o tipo a se empregar, pode ocasionar maiores respingos, maior instabilidade e grandes variações na tensão ou na corrente da soldagem e numa obtenção do cordão resultante irregular (MODENESI, 2011b, p. 62).

... a operação de soldagem causa alterações localizadas e bruscas no material sendo soldado. Estas mudanças, por sua vez, podem provocar mudanças estruturais e, conseqüentemente nas propriedades do material. Em geral estas alterações se dão na forma de uma degradação nas propriedades, o que pode ter importantes implicações na futura utilização da peça soldada. Existem duas maneiras de se enfrentar este problema. A primeira é desenvolver materiais que sejam menos sensíveis à soldagem, isto é, melhorar a *soldabilidade* dos materiais. A segunda é **controlar a operação da soldagem**² (e, possivelmente, executar operações complementares) de modo a minimizar, ou remover a degradação das propriedades da peça (MODENESI, 2011c, p. 1.13).

O sistema proposto vai manter o bico de contato numa posição relativamente fixa ao longo do processo de soldagem, corrigindo em certa medida, a posição da tocha caso aconteça algum tipo de distorção que desloque o bico de contato e possa apagar o arco ou perturbar o comprimento do arco elétrico. Estas distorções segundo Kou (2003, p. 126) são produzidas pelo aquecimento das chapas metálicas no processo e deslocam partes das chapas metálicas em diversas direções, podendo produzir variações na distância entre as peças e o bico de contato da tocha de soldagem.

² A parte destacada não é do original.

Figura 5.1.1. Esquema da distorção das peças metálicas ao longo do processo de soldagem que se tentará corrigir.



Fonte: Próprio autor.

O esquema da Fig. 5.1.1 mostra as perturbações esperadas na trajetória da soldagem e na posição do bico de contato. Com o aquecimento das peças estas tendem a se distorcer, dependendo das tensões residuais no processo de resfriamento da peça (KOU, 2003, p. 123-126).

Para medir diferenças geométricas na trajetória do sistema de soldagem pelas deformações das peças ao longo do processo de soldagem, se propõe colocar dois sensores numa posição específica que permita medir a posição e trajetória do robô em relação com as peças.

Segundo estas definições, qualquer variação ou perturbação dessa geometria implica na detecção de uma distorção no percurso da soldagem. Acredita-se que automatizando o posicionamento da tocha e corrigindo ou diminuindo o efeito da distorção das peças vai se melhorar o processo de fabricação de soldagem, acelerando a produção das peças terminadas com uma melhora na qualidade do cordão final **ao estabilizar a posição da tocha ao longo do processo**. Limita-se a solução do sistema proposto a trajetórias retas para a soldagem de peças metálicas para juntas do tipo T.

Então se precisa determinar o tipo de sensor a ser utilizado para medir a distância entre as peças e o robô de soldagem e calcular a geometria que permita caracterizar as mudanças ou variações da forma da peça a se soldar. O trabalho de Pimenta (2009) mostra resultados interessantes sobre a emissão de luz infravermelha do arco elétrico da soldagem MIG/MAG com intensidades que saturariam qualquer sensor desta tecnologia para sensoriamento de

distância em algum processo de soldagem do tipo MIG/MAG. Não se conhecem estudos sobre as vibrações na soldagem MIG/MAG ou que estas estejam na faixa do ultrassom, os valores conhecidos das taxas de deposição metálica para os processos MIG/MAG oscilam entre 0,1 e 500 gotas por segundo (MODENESI, 2011b, p. 62-94) muito afastado da faixa de frequências do ultrassom, que se localiza na faixa das dezenas de milhares de pulsos por segundo.

Então para a detecção e correção da trajetória foi escolhido o sensor por ultrassom US-100 com as seguintes características.

Tabela 5.1.1. Principais características do sensor US-100.

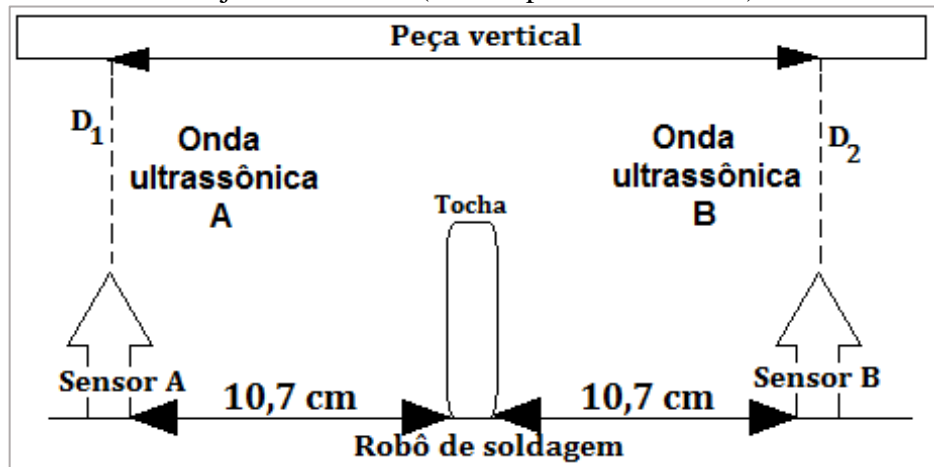
Parâmetro	Valor
Frequência	40 kHz
Faixa de medição	2 cm – 350 cm
Compensação de Temperatura	Sim
Níveis lógicos	TTL
Janela de medição	15°
Resolução	1 mm

Fonte: DTS (US-100, p. 2-3).

Dos dados da Tab. (5.1.1), avalie-se o parâmetro da frequência de emissão do sinal ultrassônico que é 40 kHz, esse valor está muito além das vibrações emitidas pelos processos de soldagem do tipo MIG/MAG, e acredita-se que se pode utilizar para medição de objetos sólidos como são as chapas metálicas de aço na soldagem MIG/MAG. Um aspecto importante na escolha do sensor US-100 é a sua característica de compensação de temperatura, dado que pela natureza da aplicação o ambiente em torno do sensor terá uma temperatura elevada.

O esquema de colocação dos sensores é apresentado na Fig. 5.1.2, sendo que o principal objetivo é traçar uma linha reta da trajetória medindo a distância entre dois pontos e monitorar ao longo do processo as variações geométricas destes pontos, qualquer variação acima do valor pré-estabelecido ativará a correção da trajetória do robô.

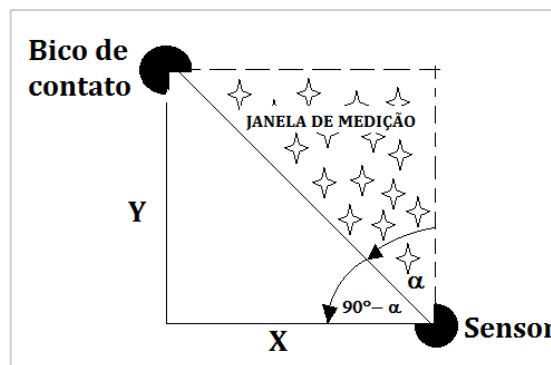
Figura 5.1.2. Posicionamento dos sensores de distância por ultrassom para correção da trajetória do robô (vista superior do sistema).



Fonte: Próprio autor.

Os sensores devem ser posicionados na mesma altura e distância afastados da tocha de soldagem procurando não detectá-la. Sabendo disso, é necessário determinar uma posição apropriada para a colocação dos sensores para resolver o problema em questão. Para determinar o posicionamento dos sensores, considere-se o dado técnico, janela de medição do US-100 que é de 15° , conhecido esse parâmetro proceder-se-á com o cálculo da distância mínima de posicionamento dos sensores.

Figura 5.1.3. Diagrama para o cálculo da distância mínima de posicionamento dos sensores US-100.



Fonte: Próprio autor.

Na Fig. 5.1.3, a variável Y é a distância entre o robô e o bico de contato durante o processo da soldagem, X é a distância de posicionamento do sensor a se encontrar, α é ângulo ou janela de medição do sensor, isto é 15° .

Pelas limitações físicas do sistema (limitações dos sensores de distância, curso útil dos sistemas CNC, comprimento da tocha de soldagem), a distância Y vai ser variável entre 8 e 20

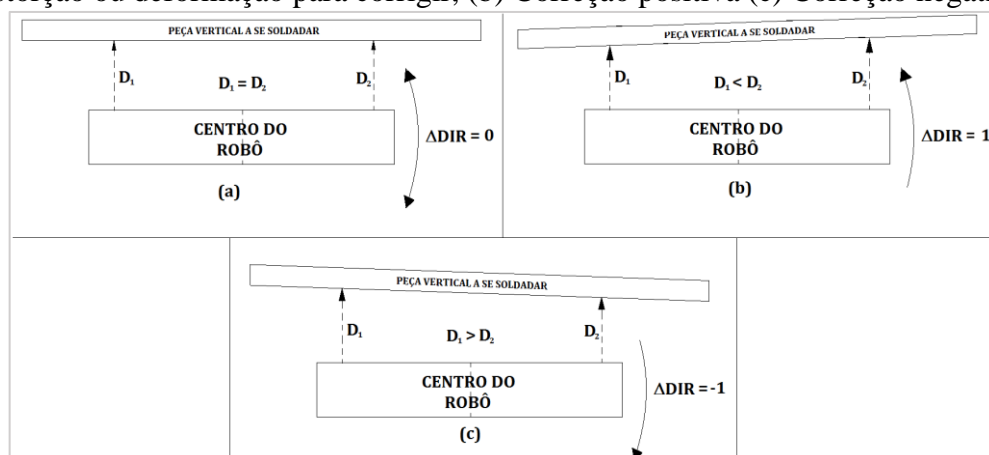
cm, considerando o pior dos casos no cálculo, Y tomará o valor de 20 cm. Com ajuda da Eq. (5.1.1) calcular-se-á distância X mínima para colocar os sensores US-100.

$$X = \frac{Y}{\tan(90^\circ - \alpha)} \quad (5.1.1)$$

Substituindo os valores correspondentes na Eq. (5.1.1), a distância X mínima para evitar erros nas medições de distância é de 5,35 cm (ambos sensores devem ser colocados na mesma distância afastados da tocha). A distância 5,35 cm é o valor mínimo de colocação, no entanto, a posição escolhida para a colocação dos sensores foi do dobro dessa distância mínima, isto assegurará uma diminuição das falsas detecções e erros nas medições. Estabeleceu-se que 10,7 cm será o valor apropriado para posicionar os sensores a ambos os lados da tocha, o resultado já foi mostrado na Fig. 5.1.2.

Antes de prosseguir com o projeto, é importante fazer uma ressalva que não existe um modelo matemático que permita aproximar matematicamente o tipo e o nível de distorção das peças a serem soldadas, no entanto pela natureza da aplicação que está se resolvendo, propôs-se a hipótese que uma variância nas distâncias D_1 e D_2 , no esquemático da Fig. 5.1.2, originaria uma variância na trajetória do bico de contato, alterando a posição e afetando o processo de soldagem. A proposta é medir a diferença destas distâncias e corrigir a trajetória, mantendo sempre (ao longo do processo de soldagem) constante a distância do bico de contato com as peças a serem soldadas. O esquema proposto apresenta-se na Fig. 5.1.4.

Figura 5.1.4. Esquema proposto para correção da trajetória e posição, (a) não há distorção ou deformação para corrigir, (b) Correção positiva (c) Correção negativa.



Fonte: Próprio autor.

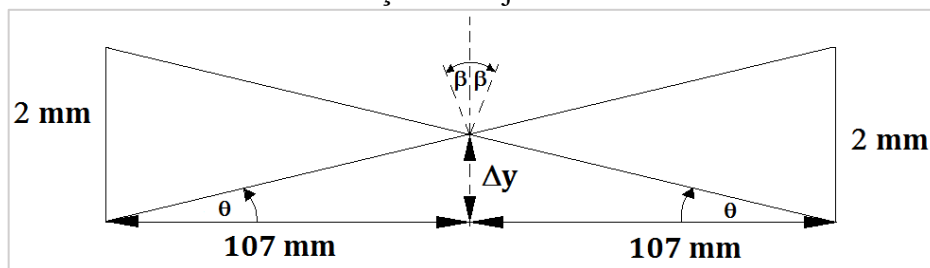
A Fig. 5.1.4 apresenta as possibilidades de correção da posição e trajetória da tocha em relação às peças. Pela resolução do sensor podem-se medir distâncias com uma precisão de 1 mm ou seja em todos os casos teremos a seguinte situação:

$$\Delta DIR = (D_1 \pm 1mm) - (D_2 \pm 1mm) = D_1 - D_2 \pm 2mm \quad (5.1.2)$$

Deste resultado extrai-se que a diferença calculada na Eq. (5.1.2), deve-se fixar num valor de 2 mm para corrigir a variância incluindo a incerteza do instrumento de medição, dito em outras palavras uma diferença medida de ± 2 mm. Acima disto fará com que o sistema aja para corrigir a posição mantendo uma distância relativamente constante entre o bico de contato e as peças.

A variação máxima do bico de contato em relação a peça é calculada a seguir (vide esquemático apresentado na Fig. 5.1.5.).

Figura 5.1.5. Variância máxima da posição do bico de contato ao detectar distorção ou deformação da trajetória linear.



Fonte: Próprio autor.

O ângulo de variação máximo permitido medido na distorção para o sistema começar corrigir é:

$$\theta = \arctan\left(\frac{2 \text{ mm}}{214 \text{ mm}}\right) \cong 0,6^\circ \quad (5.1.3)$$

A variação da posição do bico de contato em relação com as peças, segundo o esquema apresentado na Fig. 5.1.5, vai ser de:

$$\Delta y = 107 \cdot \tan(\theta) = 107 \cdot \tan(0,6) \cong 1 \text{ mm} \quad (5.1.4)$$

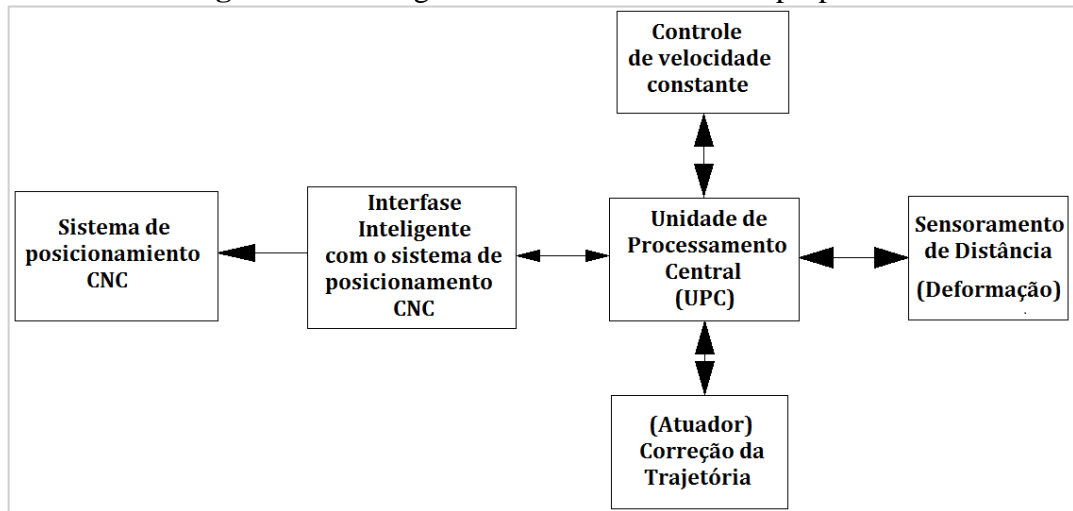
O ângulo de variação máximo do bico de contato em relação com as peças é de:

$$\beta \cong \theta \cong 0,6^\circ$$

5.2 Sistema de controle geral

Baseado nos fatos apresentados e os resultados dos cálculos projetados o esquema geral do sistema desenvolvido está o mostrado na Fig. 5.2.1.

Figura 5.2.1. Diagrama de blocos do sistema proposto.



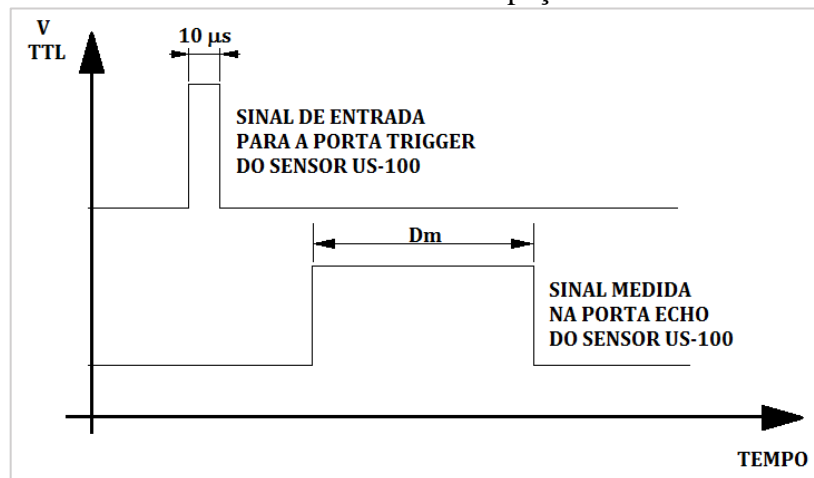
Fonte: Próprio autor.

Cada um dos blocos na Fig. 5.2.1 será projetado, explicado e apresentado nas secções seguintes, a Unidade de Processamento Central (UPC) será explicada ao final porque nela resume-se o desenvolvimento dos demais subsistemas.

5.2.1 Bloco sensoriamento da distância (Variação da trajetória)

Para projetar este subsistema é necessário analisar o funcionamento do sensor US-100, na folha de informações fornecida pelo fabricante (DTS), esse dispositivo pode ser utilizado para adquirir os dados através de um protocolo de comunicação tendo embarcada uma Unidade Assíncrona de Transmissor-Receptor (UART), para comunicação através de um protocolo de comunicação serial. Os microcontroladores disponíveis para o desenvolvimento e implantação do sistema apenas possuem uma unidade UART desse tipo, pelo que será necessário criar uma nova estrutura de comunicação para capturar as informações de distância de dois sensores para conseguir calcular a diferença das distâncias com a velocidade suficiente para corrigir a posição e o deslocamento do bico de contato e este sempre esteja dentro dos limites propostos.

Figura 5.2.2. Diagrama de temporização do sensor de distância US-100, usado para medir a distância do robô à peça.



Fonte: Modificada do DTS (US-100).

Na Fig. 5.2.2, os sinais ilustram como utilizar o sensor US-100 para medir a distância requerida para o cálculo do ΔDIR com a Eq. (5.1.2), sem a necessidade da unidade UART, este esquema foi implementado na UPC, a qual gera um sinal lógico do nível da tecnologia *Transistor-Transistor-Logic* (TTL) por um período de 10 μs , após a geração desse sinal, foi necessária a programação de uma função de espera da saída pela porta ECHO do sensor US-100, a largura temporal desse pulso de saída gerada pelo sensor é proporcional à distância medida. A fórmula para a medição da distância percorrida pela onda sonora é dependente da velocidade do som no ar seco, isto é dado pela expressão a seguir:

$$V_{el} = 331,4 + 0,6T_{amb} \left[\frac{m}{s} \right] \quad (5.2.1)$$

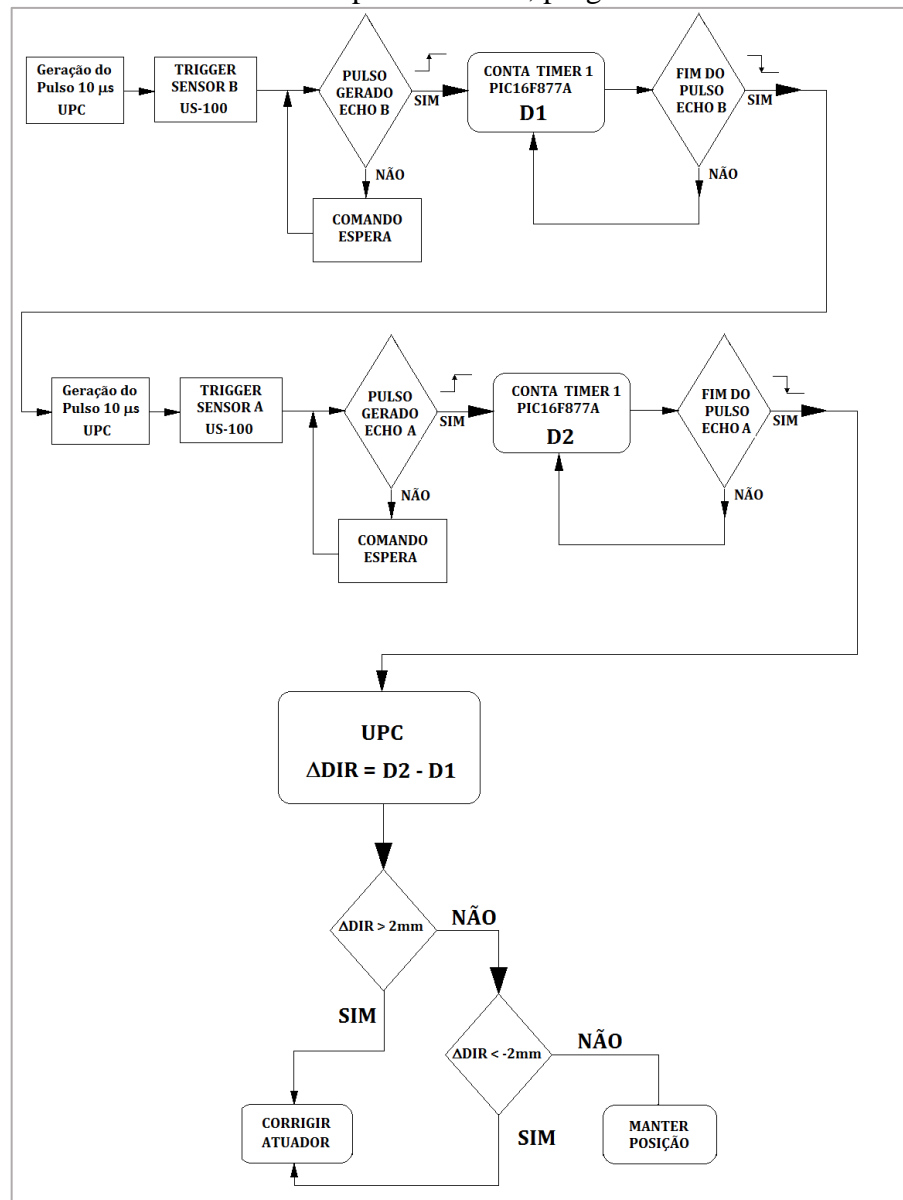
Onde V_{el} é a velocidade do som, T_{amb} é a temperatura do ambiente, é importante lembrar que o US-100 é compensado em temperatura, que para esse dispositivo a Eq. (5.2.1) resume-se a 331,4 m/s. Desta forma, e pelas características próprias do sensor, a distância medida na porta ECHO será proporcional à largura do pulso D_m da forma

$$\text{Distancia}_m = D_m \times 0,167 [mm] \quad (5.2.2)$$

Com D_m em microssegundos. Para conseguir colocar D_m em essa unidade de tempo dentro do microcontrolador, foi preciso o uso do temporizador interno configurado nesta unidade de tempo. O PIC16F877A possui internamente três temporizadores, tendo que um deles apresenta uma resolução de 16 bits, o que permitirá medir as distâncias na faixa estabelecida. A maior distância a se medir pelas limitações do sistema será 215 mm o que equivaleria a um pulso D_m de 1204 μs .

O algoritmo computacional de cálculo da UPC baseado nas medições da distância apresenta-se na Fig. 5.2.3, o resultado desse processo de cálculo será a entrada do subsistema atuador ou correção da trajetória.

Figura 5.2.3. Algoritmo computacional para o cálculo da correção na trajetória e posição baseado em sensoriamento por ultrassom, programado no PIC16F877A.



Fonte: Próprio autor.

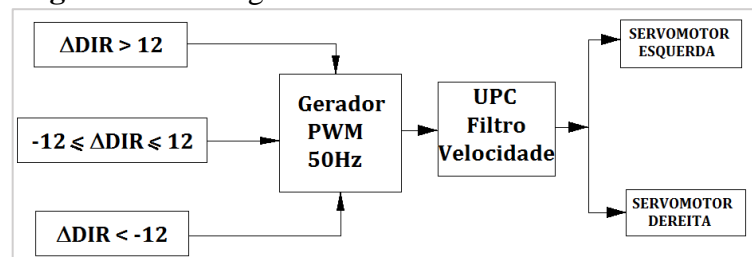
5.2.2 Bloco atuador (corretor da trajetória)

O sistema atuador vai agir dependendo do resultado do subsistema sensorial da distância por ultrassom, sendo que esta correção é também dependente da velocidade de

deslocamento do sistema, por isto é necessário que a velocidade mantenha-se relativamente constante para que a correção na trajetória possa ser efetiva.

Desde o ponto de vista mecatrônico foi definido como atuador um par de servomotores trabalhando em paralelo, obtendo como resultante um mecanismo de tamanho compacto e com a força necessária para manejar as duas rodas. As peças mecânicas deste protótipo foram feitas manualmente com uma retifica DREMEL TRIO[®], pelo que o mecanismo apresentará esforços maiores dos calculados. O diagrama de blocos deste subsistema apresenta-se na Fig. 5.2.4.

Figura 5.2.4. Diagrama de blocos do subsistema atuador.



Fonte: Próprio autor.

O servomotor como foi pesquisado é um dispositivo empregado para atuar sobre um eixo e conseguir uma posição angular entre -90° e 90° de forma precisa. Embora a faixa dos servos-motores seja de 180° , para esta aplicação não será necessária a utilização de todo o intervalo mencionado.

O principal objetivo deste subsistema é manter a trajetória numa linha reta paralela à peça vertical da junta tipo T, e corrigir as diferenças medidas no caso a peça apresente uma mudança dessa trajetória reta mantendo a posição da tocha relativamente constante.

Como visto na Fig. 5.1.4, o ângulo de variação máximo calculado com a Eq. (5.1.3), antes de começar corrigir a trajetória vai ser de $0,6^\circ$, mas este ângulo é estático e absoluto. Para corrigir essa variação após a detecção dos sensores de distância e o correspondente cálculo por parte do microcontrolador, é necessário considerar a velocidade de deslocamento do sistema na direção da produção do cordão de solda.

A Eq. (5.2.3) é característica do sensor de velocidade empregado, obtida a partir do conceito básico de velocidade da física geral, segundo Serway (2005, p. 29).

A função velocidade linear do robô em termos da posição e o tempo é dada pela equação a seguir:

$$v(t) = \frac{dx}{dt} = \frac{\Delta x}{\Delta t} = \frac{55,56}{t_p} \left[\frac{cm}{min} \right] \quad (5.2.3)$$

No tempo zero quando é enviada a ordem de correção o servo vai demorar um tempo até se posicionar no ângulo α para correção. Neste ponto podemos expressar a velocidade angular de correção com a Eq. (5.2.4), (SERWAY, 2005, p.296).

$$\omega = \omega_0 + \alpha \cdot t \quad (5.2.4)$$

Considerando a característica de operação dos servos-motores utilizados que foram os TowerPro SG90 a velocidade inicial $\omega_0 = 0,6/t_p^2$, isto é 60° vezes $0,1$ s, em relação com o tempo de amostragem (que é o mesmo tempo de resposta do sensor).

Dos parâmetros conhecidos do projeto, no momento exato do início da correção $\theta_0 = -0,6^\circ$ e o ângulo final $\theta_f = 0^\circ$ e para a maior das velocidades $t_p \approx 0,5$ s.

$$\frac{\Delta\theta}{t_p} = \frac{0,6}{t_p^2} + \alpha \cdot t_p \quad (5.2.5)$$

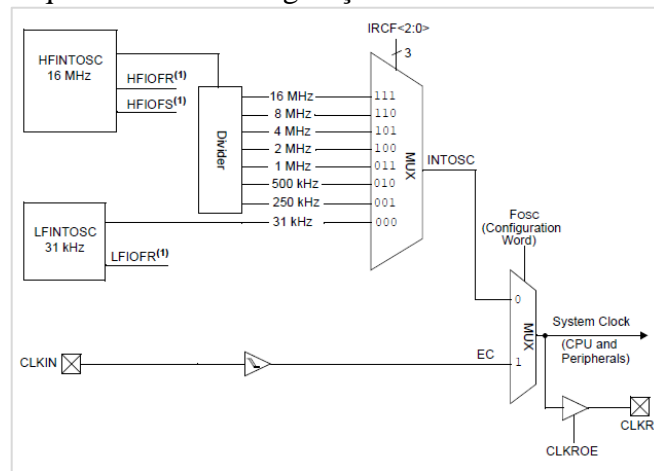
$$\frac{-0,6}{0,5} = \frac{0,6}{0,25} + \alpha \cdot 0,5 \quad (5.2.6)$$

$$\alpha \approx -7,2^\circ$$

A interpretação final deste resultado é que mudando o ângulo das rodas $7,2^\circ$ na direção oposta ao movimento é suficiente para corrigir a direção com ajuda do servomotor SG90 para qualquer velocidade de soldagem, usando o tempo de amostragem do encoder. *No projeto mecânico o ângulo final das rodas foi de aproximadamente $\pm 15^\circ$.*

Para o controle do servomotor foi utilizado o microcontrolador PIC10F322, o qual possui internamente um oscilador configurável em várias frequências. Esta característica era necessária devido a que o servomotor para sua correta operação deve receber um pulso a cada 50 Hz, a largura deste pulso é a que vai definir o ângulo final da alavanca do servomotor. Essa frequência tão baixa é incompatível com as frequências configuradas com os demais sistemas tanto no PIC16F877A quanto no PIC18F4550.

Figura 5.2.5. Esquemático da configuração do oscilador interno do PIC10F322.



Fonte: DTS (PIC10F322).

A frequência escolhida para o sistema atuador foi de 2 MHz, tendo que esta frequência é necessária para a configuração do período do PWM. Para fixar este período no valor certo, é necessário definir na programação do microcontrolador o pré-escalador do TIMER 2 com valor igual a 64, o registro interno PR2 com valor igual a 155, a fórmula e configuração para obter a frequência baseado nestes valores dos registros internos é dada pela equação a seguir, (DTS, PIC10F322).

$$f_{PWM} = \frac{f_{osc}}{4 \cdot (PR2 + 1)(PRE_TMR2)} = \frac{2MHz}{4 \cdot (156)(64)} \quad (5.2.7)$$

$$f_{PWM} \cong 50Hz$$

A resolução do PWM para este microcontrolador é dada pela Eq. (5.2.8), (DTS, PIC10F322).

$$Resolução = \frac{\log[4(PR2 + 1)]}{\log(2)} = \frac{\log[624]}{\log(2)} \quad (5.2.8)$$

$$Resolução = 9,2 \text{ bits}$$

Quer dizer que para um ciclo de trabalho do 100% o número a carregar será 588, a Tab. (5.2.1) definirá os valores a se usar para o PWM dos servos.

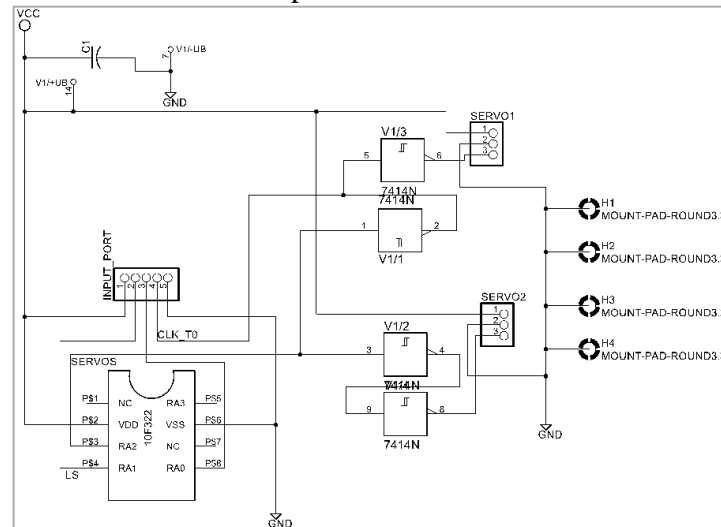
Tabela 5.2.1. Valores utilizados do Ciclo de Trabalho para os servos-motores.

Ângulo	Valor (inteiro 16 bits)	Ciclo de Trabalho (%)	Tempo em alto (ms)
-15°	48	8,1	1,62
0	54	9,1	1,82
15°	59	10,0	2

Fonte: Próprio autor.

O circuito resultante montado no Cadsoft Eagle V5.0 é apresentado na Fig. 5.2.6.

Figura 5.2.6. Circuito eletrônico para o sistema atuador de controle dos servos.

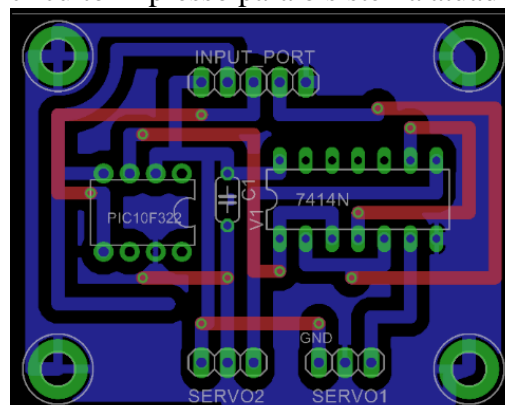


Fonte: Próprio autor.

No subsistema de controle dos servos-motores colocou-se um dispositivo auxiliar que foi o circuito integrado 7414N que inclui 6 amplificadores com Schmitt-Triger, o qual pode ser usado para não carregar o microcontrolador e evitar que o ruído elétrico gerado pelos motores dos servos for direto para o microcontrolador. O layout do circuito eletrônico da Fig. 5.2.6 apresenta-se na Fig. 5.2.7.

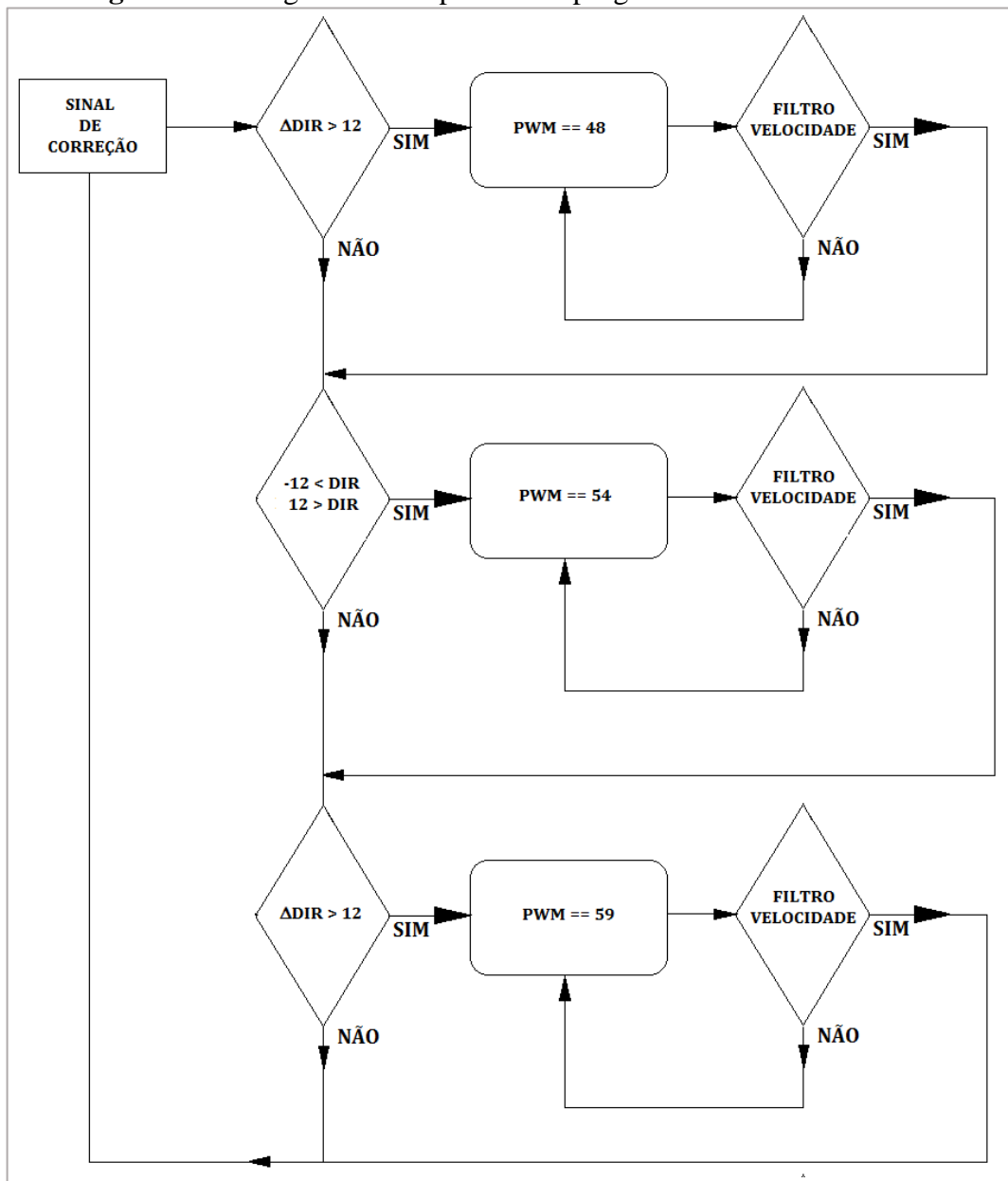
O fluxograma da Fig. 5.2.8 detalha o algoritmo computacional programado no microcontrolador PIC10F322 para o controle dos servos-motores.

Figura 5.2.7. Layout do circuito impresso para o sistema atuador de controle dos servos.



Fonte: Próprio autor.

Figura 5.2.8. Algoritmo computacional programado no PIC10F322.



Fonte: Próprio autor.

5.2.3 Bloco Controle da Velocidade Constante

5.2.3.1 Modelagem do motor de corrente contínua para o controle da velocidade

Esta secção foi feita baseada nos trabalhos de Pichardo (2008), Quijano e Hernandez (2009), para a obtenção dos parâmetros do motor de corrente contínua e a modelagem do

motor no domínio da frequência, a Eq. (5.2.9) até a Eq. (5.2.15), foram compiladas do trabalho de Quijano e Hernandez (2009).

5.2.3.2 Constante contraeletromotriz

A constante contraeletromotriz pode ser calculada com ajuda da Eq. (5.2.9), medindo a tensão e a velocidade do motor simultaneamente.

$$K_e = \frac{E_b}{n} = \frac{V}{rad/s} \quad (5.2.9)$$

O resumo das medições feitas para o cálculo desta constante se apresenta na Tab. (5.2.2), o valor da constante encontrada é a média baseado nos resultados de múltiplas velocidades.

Tabela 5.2.2. Medições de velocidade do motor para o cálculo da constante K_e .

$E_b [V]$	$n [rpm]$	$K_e [V/rpm]$	$K_e [\frac{V}{rad/s}]$
2,14	1,18	1,82	0,1908
2,52	1,40	1,8	0,1887
3,10	1,74	1,78	0,1866
4,95	2,84	1,75	0,1834
5,38	3,11	1,73	0,1813
6,33	3,60	1,76	0,1844
6,83	3,93	1,74	0,1823
7,25	4,10	1,77	0,1855
8,45	4,44	1,90	0,1991
9,00	5,00	1,8	0,1887
9,41	5,21	1,81	0,1897
10,5	5,93	1,77	0,1855
11,01	6,00	1,84	0,1928
11,75	6,10	1,93	0,2023
12,12	6,20	1,95	0,2044
Médias		1,81	0,1898

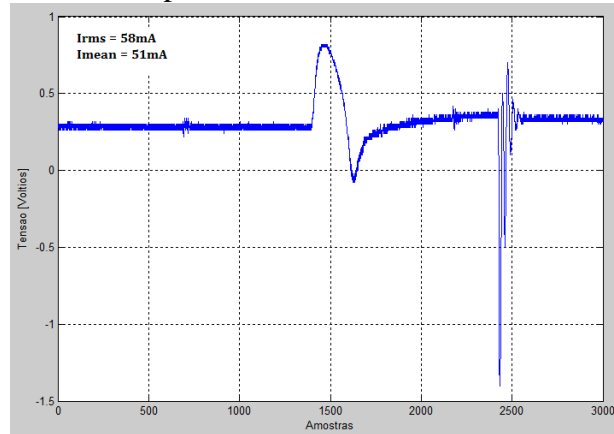
Fonte: Próprio autor.

5.2.3.3 Resistência da armadura

A resistência da armadura do motor de corrente contínua pode ser determinada através da medição da tensão e a corrente no motor antes de começar girar. Com estes valores aplica-se a lei de Ohm (ALEXANDER e SADIKU, 2006, p. 30).

$$R_a = \frac{V}{I} = \frac{1,4V}{58mA} = 24\Omega \quad (5.2.10)$$

Figura 5.2.9. Corrente de rompimento da zona morta no motor de corrente contínua.



Fonte: Próprio autor.

A Fig. 5.2.9 apresenta a medição da corrente ao aplicar uma tensão de aproximadamente 1,4V nos terminais do motor, dita medição foi tomada com o osciloscópio no modo de aquisição, foram tomadas 3000 amostras e o cálculo da tensão RMS foi feita no programa Octave. Tendo medido essa corrente, o cálculo da resistência da armadura do motor foi feito usando a Eq. (5.2.10).

5.2.3.4 Indutância da armadura.

A indutância da armadura foi encontrada com ajuda do instrumento de medição LCR, o valor obtido é apresentado na Fig. 5.2.10.

$$L_a = 1,765 \text{ mH} \quad (5.2.11)$$

Figura 5.2.10. Medição da indutância do motor com o LCR.



Fonte: Próprio Autor.

5.2.3.5 Constante de tempo elétrica

O cálculo deste parâmetro foi feito de forma indireta, conhecendo os valores da indutância e a resistência da armadura, o parâmetro em questão pode ser obtido através da Eq. (5.2.12).

$$t_e = \frac{L_a}{R_a} = \frac{1,765 \text{ mH}}{24\Omega} = 73,5 \mu\text{s} \quad (5.2.12)$$

5.2.3.6 Constante do torque do motor

A constante do torque, pode ser encontrada a partir da constante contraeletromotriz, arranjando as unidades e multiplicando pelo fator 95.493 rad/krpm , as unidades resultantes da constante serão $[\text{Nm/A}]$.

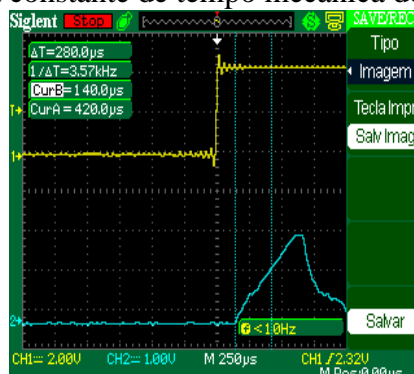
$$K_t = 95.493 \cdot K_e \left[\frac{\text{V}}{\text{krpm}} \right] \quad (5.2.13)$$

$$K_t = 0,1718 [\text{Nm/A}]$$

5.2.3.7 Constante de tempo mecânica

Para poder medir a constante de tempo mecânica, foi necessário ajustar a tensão nominal do motor em 12 V e aplicar esse sinal com uma frequência que se permitirá enxergar a carga e descarga da corrente do motor, uma vez feito esse ajuste é necessário medir o nível de corrente do motor quando atingir o 63,2%.

Figura 5.2.11. Medição da constante de tempo mecânica do motor com o osciloscópio.



Fonte: Próprio autor.

O valor medido é observado na Fig. 5.2.11 e é mostrado a seguir.

$$t_m = 280\mu s \quad (5.2.14)$$

5.2.3.8 Momento de inércia do motor CC.

O momento de inércia pode ser calculado de forma indireta com a Eq. (5.2.15), empregando os valores obtidos dos parâmetros anteriores.

$$J_m = \frac{(t_m)(K_t)(K_e)}{R_a} = \frac{(s)(Nm/A)(V/(rad - s))}{(\Omega)} = [kg - m^2] \quad (5.2.15)$$

$$J_m = \frac{(280)(0,1718)(0,1887)}{24} = 3,78 \times 10^{-7} [kg - m^2]$$

5.2.3.9 Atrito do rotor no motor

O parâmetro de viscosidade, que produz uma força oposta ao movimento do rotor do motor, esse fenômeno físico devido ao esforço inercial deve ser ultrapassado para o eixo se movimentar. Para encontrar a constante de viscosidade é necessário traçar um gráfico, com ajuda das Eq. (5.2.15) e (5.2.16) que representam o torque necessário para vencer esse atrito.

$$T_f = K_t I_{START} \quad (5.2.16)$$

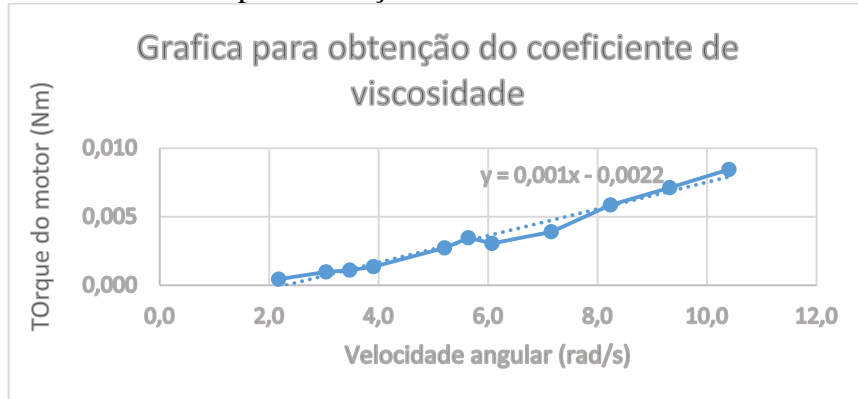
$$T_f = (0,1887 Nm/A)(3.85A) = 0,6614 Nm$$

5.2.3.10 Constante de Viscosidade de Coulomb

$$T_m = K_t I_a = B\omega + T_f \quad (5.2.17)$$

O traço da Eq. (5.2.17) é apresentado no gráfico da Fig. 5.2.12, a inclinação desta curva é a constante de viscosidade B.

Figura 5.2.12. Gráfico para obtenção do coeficiente de viscosidade mecânica.



Fonte: Próprio autor.

5.2.3.11 Função de transferência do motor CC

Finalmente a função de transferência do motor baseado em todos os parâmetros encontrados pode ser descrita a partir da Eq. (5.2.18)

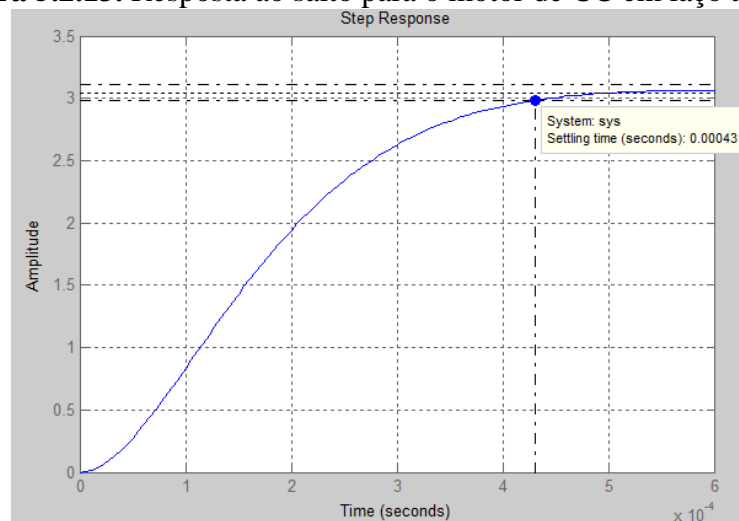
$$\frac{\omega}{V} = \frac{K_t}{(sL_a + R_a)(sJ_m + B) + K_t K_e} \quad (5.2.18)$$

Substituindo os valores obtidos das constantes do motor chega-se à função de transferência do sistema que se mostra a seguir.

$$H(s) = \frac{288,8 \times 10^6}{s^2 + 16565 \cdot s + 94,9 \times 10^6}$$

A resposta ao salto dessa função de transferência apresenta-se na Fig. 5.2.13

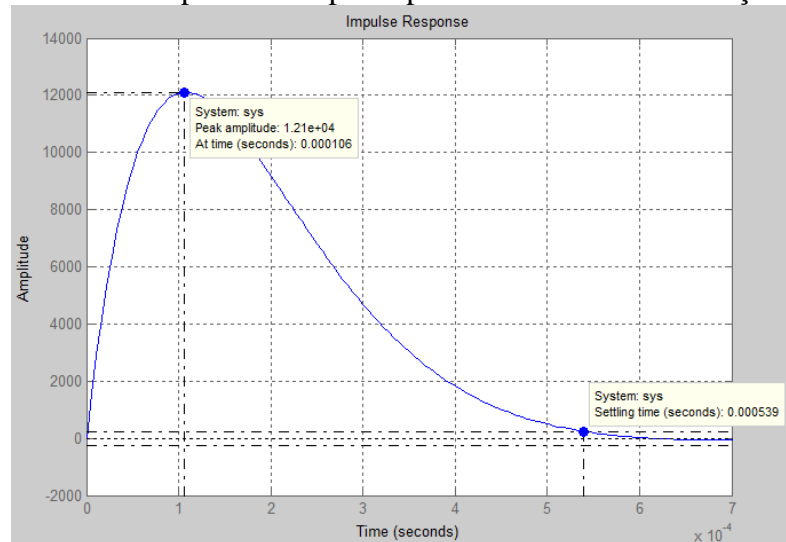
Figura 5.2.13. Resposta ao salto para o motor de CC em laço aberto.



Fonte: Próprio autor.

No gráfico da Fig. 5.2.13 o tempo de estabilização ($T_{setting}$) obtido para o motor foi de $431\mu s$. A resposta ao impulso do motor apresenta-se na Fig. 5.2.14.

Figura 5.2.14. Resposta ao impulso para o motor de CC em laço aberto.



Fonte: Próprio autor.

No gráfico da Fig. 5.2.14 o tempo de pico do sinal obtido para o motor foi de $106\mu s$ e o tempo de estabilização dessa resposta foi de $539\mu s$.

A frequência de amostragem (inversamente proporcional ao tempo de amostragem) mínima para o controle da velocidade do motor deve ser menor do que o tempo de estabilização da resposta ao salto (INTERIANO, 2010, p. 102).

$$T_s \leq \frac{T_{setting}}{10} \quad (5.2.19)$$

$$T_s \leq \frac{431\mu s}{10} = 43,1\mu s$$

O período estabelecido para o PWM e que será aplicado no motor de CC, deverá ser menor do que o tempo (T_s) obtido, a fim de garantir que a propriedade eletromagnética que produz o torque do motor não veja o PWM como uma perturbação elétrica. No caso da resposta ao impulso o valor pico do sinal é dado em $106\mu s$

$$T_s \geq \frac{106\mu s}{10} = 10,6\mu s \quad (5.2.20)$$

Para evitar perturbações impulsivas o patamar do período do PWM será então

$$10,6\mu s \leq T_s \leq 43,1 \mu s \quad (5.2.21)$$

5.2.3.11 Sistema com atraso devido ao sensor de velocidade

A função de transferência do motor incluindo o atraso do sensor pode ser expressa da forma (INTERIANO, 2010, p. 12).

$$H(s) = \frac{(288,8 \times 10^6)e^{-s \cdot t_{delay}}}{s^2 + 16525 \cdot s + 94,9 \times 10^6} \quad (5.2.22)$$

Esse atraso da função em laço aberto do sinal é dado pelo encoder empregado para medir a velocidade do sistema, o retardo depende da velocidade e é dado pela expressão a seguir.

$$t_{delay} = \frac{55,56}{Velocidade[cm/min]} [s] \quad (5.2.23)$$

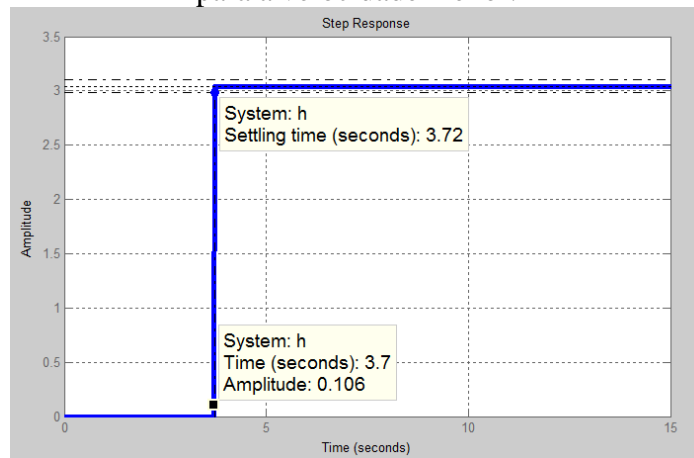
Tabela 5.2.3. Atrasos nas medições de velocidade no encoder.

Velocidade [cm/min]	t_{delay} [s]
15	3,7
110	0,5

Fonte: Próprio autor.

A seguir são apresentadas algumas respostas em laço aberto do sistema modelado incluindo o atraso do encoder na medição. Nessas respostas enxerga-se que o tempo de atraso predominante é muito próximo aos tempos expostos na Tab. (5.2.3) quando o sensor é utilizado.

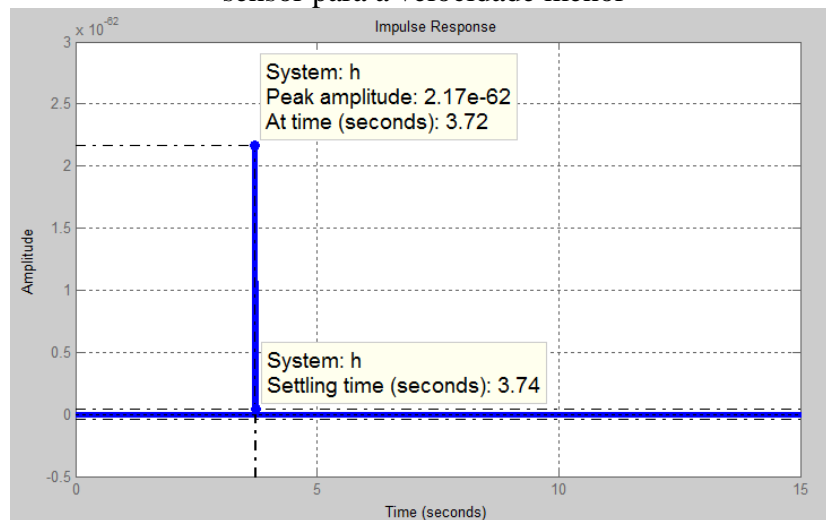
Figura 5.2.15. Resposta ao salto em laço aberto para o motor de CC com o atraso do sensor para a velocidade menor.



Fonte: Próprio autor

No gráfico da Fig. 5.2.15 enxerga-se o tempo de atraso na resposta ao salto do motor que é 3,72 s, esse atraso é praticamente o mesmo tempo de estabilização do sistema, que corresponde aos atrasos vistos na Tab. (5.2.3).

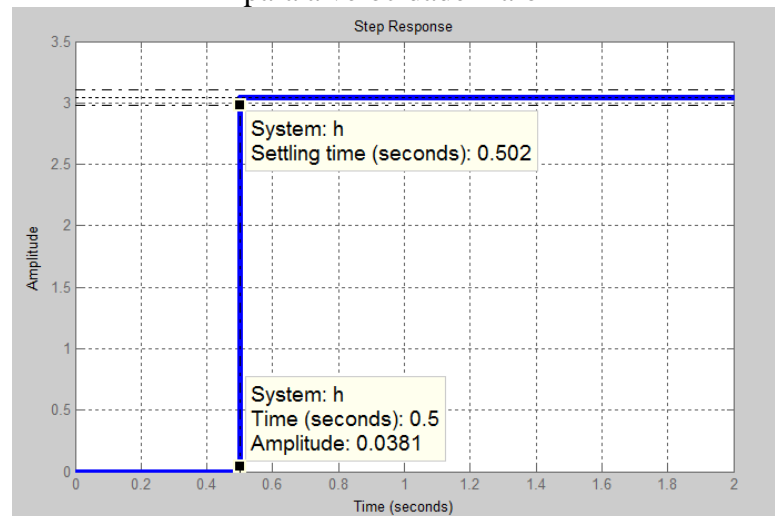
Figura 5.2.16. Resposta ao impulso em laço aberto para o motor de CC com o atraso do sensor para a velocidade menor



Fonte: Próprio autor

Na Fig. 5.2.16 o tempo de atraso da resposta ao impulso é de 3,72 s, e o tempo de estabilização é aproximadamente de 3,74 s, usando o sensor para medir a velocidade no sistema.

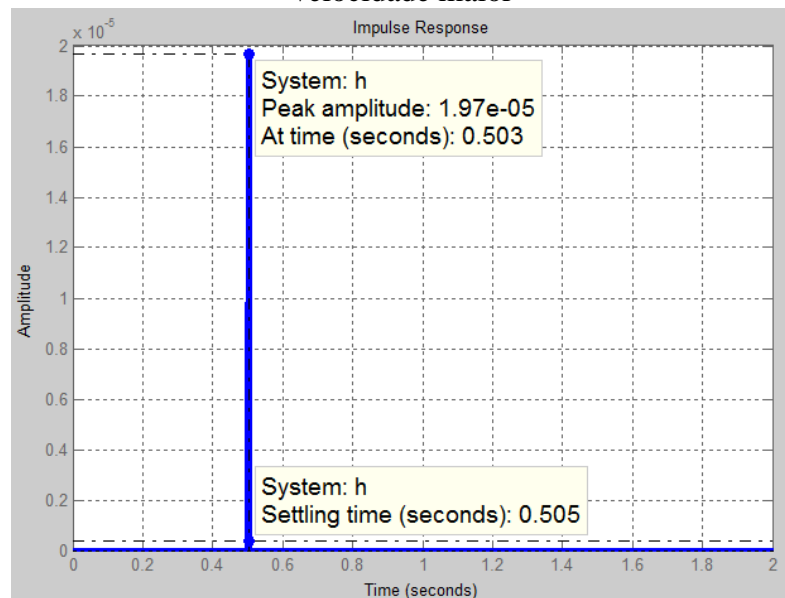
Figura 5.2.17. Resposta ao salto em laço aberto para o motor de CC com o atraso do sensor para a velocidade maior



Fonte: Próprio autor

No gráfico da Fig. 5.2.17 o tempo de atraso da resposta ao salto para a velocidade de 110 cm/min é de 0,5 s.

Figura 5.2.18. Resposta ao impulso para o motor de CC com o atraso do sensor para a velocidade maior



Fonte: Próprio autor

Da mesma forma no gráfico da Fig. 5.2.18 o tempo de estabilização para essa velocidade máxima foi próximo ao tempo de atraso nas medições da velocidade que é 0,503 s.

Uma proposta para evitar que o atraso comprometa o controle da velocidade e assim permitir ao sistema reagir às perturbações é monitorar a corrente do motor (a variação na

velocidade do motor é proporcional à variação da sua corrente). O controle da velocidade e os ajustes finais na velocidade do motor deverão se fazer empregando o sensor tipo Encoder escolhido para ter alta precisão na velocidade de soldagem.

O Encoder utilizado possui internamente 10 divisões que geram 10 pulsos aproximadamente iguais ao dar uma volta completa o que permitirá conhecer a distância percorrida em cada divisão (ou em cada pulso). O controle mais apropriado para este tipo de aplicações é o Proporcional-Integrativo (PI) (PICHARDO e CENTENO, 2008, p. 33), no entanto em sistema com grandes atrasos como é o caso do encoder, para melhorar o controle do sistema o controle PI será integrado em paralelo com o gerador do ciclo de trabalho do PWM.

5.2.3.12 Estratégia do controle do motor de CC sem atraso na medição

A estratégia a seguir vai focar-se nas anotações expostas na Tab. (5.2.4) considerando que não existe atraso nas medições da velocidade.

Tabela 5.2.4. Parâmetros de projeto do sistema de controle de velocidade

Tempo de estabilização máximo (s)	Sobressalto máximo (%)
4	25

Fonte: Próprio autor

Primeiramente vai se traçar a resposta em laço fechado da função de transferência do motor com um ganho proporcional de 1, a que é dada pela Eq. (5.2.24), (KUO, 1996, p. 83)

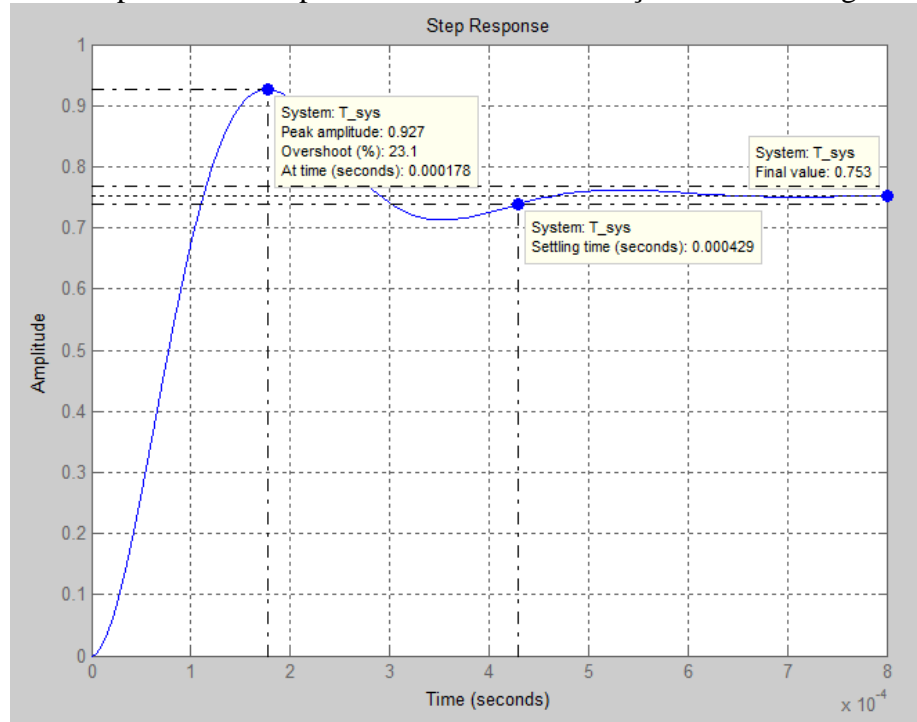
$$G(s) = \frac{K_p \cdot H(s)}{1 + K_p \cdot H(s)} \quad (5.2.24)$$

$$G(s) = \frac{288,8 \times 10^6}{s^2 + 16565 \cdot s + 383,7 \times 10^6}$$

No gráfico da Fig. 5.2.19, em laço fechado o motor já com um ganho K_p unitário, fica próximo nos parâmetros escolhidos, o tempo de estabilização obtido foi de 429 μ s. Basicamente o tempo de estabilização de 4 s, está sendo fixado devido ao intervalo temporal considerado apropriado no processo de soldagem sem importar a velocidade do processo. A resposta em laço fechado desse sinal visto na Fig. 5.2.19 apresenta sobressalto de 23,1%.

O motor em laço fechado apresenta um erro de estado estacionário de aproximadamente 24,7%, no entanto isto é ajustável no gerador do PWM no microcontrolador, então isto não será um aspecto importante no projeto do filtro de controle PI.

Figura 5.2.19. Resposta ao salto para o motor de CC em laço fechado com ganância unitária.



Fonte: Próprio autor.

5.2.3.13 Filtro de controle PI

O controle Proporcional-Integrativo apresenta uma função de transferência da forma (KUO, 1996, p. 691).

$$C_{PI}(s) = K_p + \frac{K_i}{s} \quad (5.2.25)$$

A sintonização do filtro de controle do motor no laço fechado foi feita usando a ferramenta de software Matlab. As constantes obtidas são mostradas na Tab. (5.2.5). A Eq. (5.2.26) mostra um filtro resultante do tipo integrativo puro (I).

Tabela 5.2.5. Constantes de compensação para controle de perturbações do motor em tempo contínuo.

K_p	K_i
0	600

Fonte: Próprio autor.

$$C_{PI}(s) = \frac{600}{s} \quad (5.2.26)$$

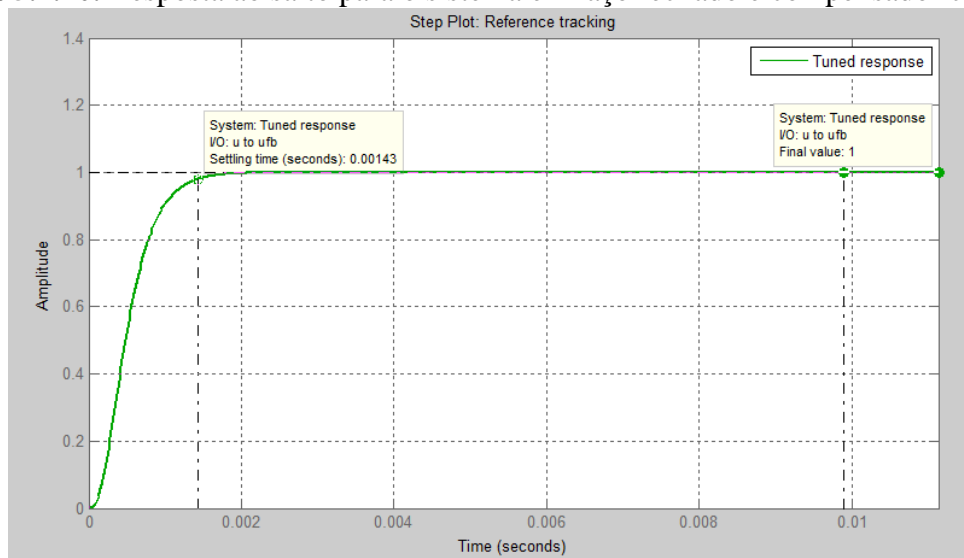
A função de transferência do sistema em laço fechado com o filtro incorporado é dada pela Eq. (5.2.27)

$$G_{PI}(s) = \frac{C_{PI}(s) \cdot H(s)}{1 + C_{PI}(s) \cdot H(s)} \quad (5.2.27)$$

$$G_{PI}(s) = \frac{173,3 \times 10^9}{s^3 + 16565 \cdot s^2 + 94,9 \times 10^6 \cdot s + 173,3 \times 10^9}$$

A resposta ao salto do sistema da Eq. (5.2.27) com o filtro integrativo puro resultante da Eq. (5.2.26) em laço fechado é apresentada na Fig. 5.2.20. Os principais dados obtidos da Fig. 5.2.20 são apresentados na Tab. (5.2.6).

Figura 5.2.20. Resposta ao salto para o sistema em laço fechado e compensador contínuo.



Fonte: próprio autor.

Tabela 5.2.6. Resumo da resposta ao salto do sistema compensado da Fig. 5.2.20.

Tempo de assentamento (ms)	1,43
Tempo de levantamento (μs)	784
Sobre-salto (%)	0
Valor final	1

Fonte: Próprio autor.

5.2.3.14 Compensador em tempo discreto.

O método empregado para encontrar a função de compensação em tempo discreto é a transformada bilinear de Tustín que é da forma (INTERIANO, 2012, p. 116).

$$s \cong \frac{2z - 1}{Tz + 1} \quad (5.2.28)$$

O tempo de amostragem utilizado para a transformação é $T = 300 \mu s$

$$C_I(z) = \left(\frac{0,09z - 0,09}{z - 1} \right) = \left(\frac{0,09 - 0,09 \cdot z^{-1}}{1 - z^{-1}} \right) \quad (5.2.29)$$

$$C_I(z) = \frac{0,09 - 0,09 \cdot z^{-1}}{1 - z^{-1}} \quad (5.2.30)$$

Fazendo uso da série de potências do Mc Laurin pode-se obter a equação de diferenças, tendo que é conhecida a série de potências das funções que conformam $C_I(z)$ (ALVARADO, 2008, p. 58).

$$\frac{1}{1 - x} = 1 + x + x^2 + \dots + x^n \quad (5.2.31)$$

Fazendo uma substituição de variável da forma:

$$x = z^{-1} \quad (5.2.32)$$

$$\frac{1}{1 - z^{-1}} = 1 + z^{-1} + z^{-2} + \dots + z^{-n} \quad (5.2.33)$$

Para garantir a estabilidade do sistema $|z^{-1}| \ll 1$ ((ALVARADO, 2008, p. 187). Tendo sempre verdadeira essa condição a série pode ser aproximada como:

$$\frac{1}{1 - z^{-1}} \cong 1 + z^{-1} \quad (5.2.34)$$

Desta forma a função de transferência pode mudar substituindo o resultado anterior nela

$$C_I(z) = (0,09 - 0,09 \cdot z^{-1}) \left(\frac{1}{1 - z^{-1}} \right) \quad (5.2.35)$$

$$C_I(z) = (0,09 - 0,09 \cdot z^{-1})(1 + z^{-1}) \quad (5.2.36)$$

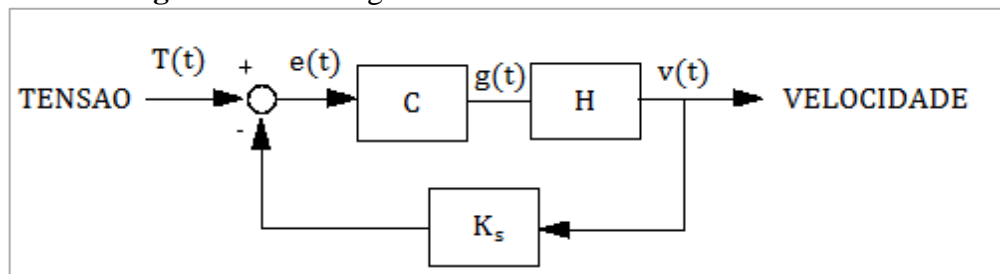
$$C_I(z) = (0,09 - 0,09 \cdot z^{-1} + 0,09 \cdot z^{-1} - 0,09 \cdot z^{-2}) \quad (5.2.37)$$

De novo pela condição de estabilidade a equação anterior pode ser aproximada por:

$$C(z) \cong 0,09$$

O controlador resultante foi simplificado utilizando o critério de estabilidade dos sistemas discretos $|z^{-1}| \ll 1$, segundo (ALVARADO, 2008, p. 187).

Figura 5.2.21. Diagrama de blocos do motor com feedback.



Fonte: Modificado de Kuo (1996, p. 83-86.).

A equação de diferenças pode ser obtida a partir da função anterior com $K_s = 1$

$$\frac{G(z)}{E(z)} = 0,09 \quad (5.2.38)$$

$$G(z) = 0,09 \cdot E(z) \quad (5.2.39)$$

Finalmente fazemos a transformada inversa (KUO, 1996, p. 64) da função anterior para obter a equação de diferenças no domínio do tempo discreto.

$$g(k) = 0,09 \cdot e(k) \quad (5.2.40)$$

Tendo que $g(k)$ é a função de compensação no domínio do tempo discreto, é equivalente à diferença das amostras atual e anterior (dentro do microcontrolador), quando a função erro seja zero a compensação do sistema será zero pelo tanto:

$$g(k) = m(k) - m(k - 1) \quad (5.2.41)$$

Sendo que $m(k)$ representa a amostra atual e $m(k-1)$ representa a amostra anterior (dentro do microcontrolador)

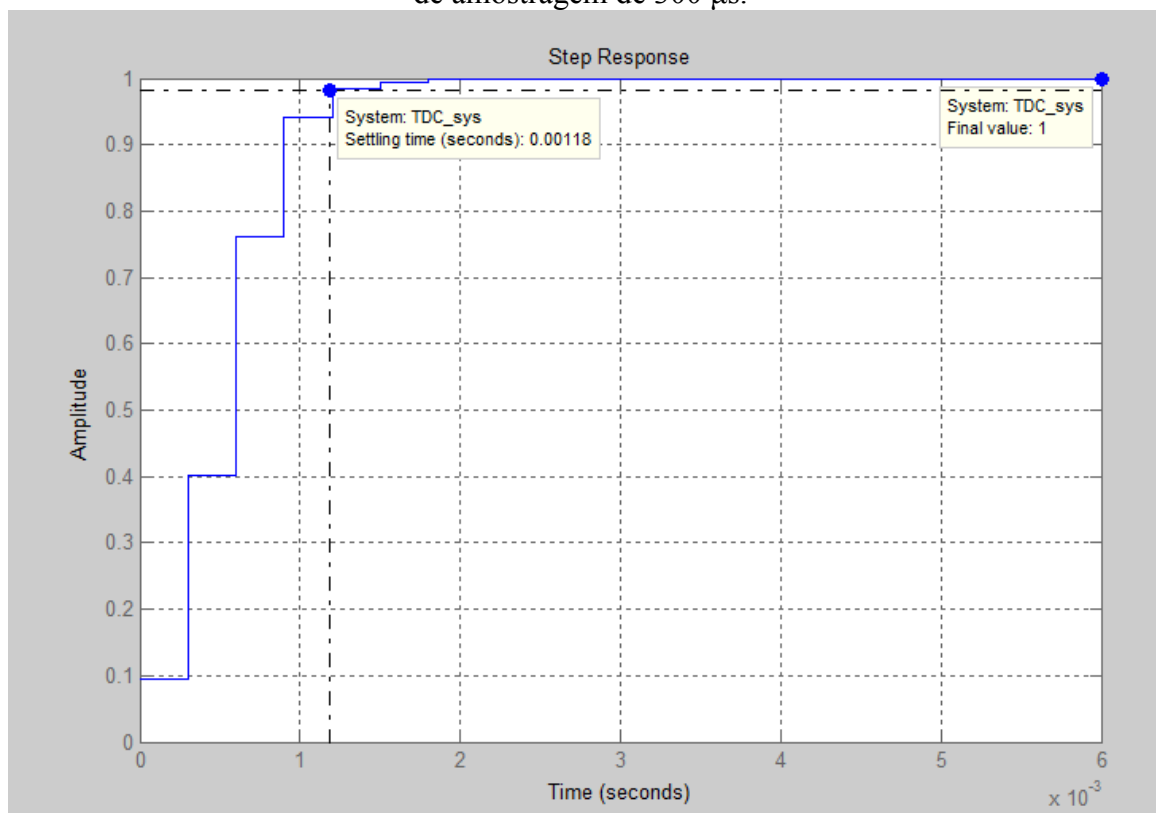
$$m(k) - m(k - 1) = 0,09 \cdot e(k)$$

$$m(k) = 0,09 \cdot e(k) + m(k - 1) \quad (5.2.42)$$

A Eq. (5.2.42) anterior pode ser utilizada para compensar as perturbações do motor (utilizando um microcontrolador) baseado nas variações da corrente da armadura que é proporcional às variações do torque da carga e assim à velocidade do motor, devido às incertezas dos modelos matemáticos ao respeito dos sistemas físicos, a função obtida apenas servirá para compensar essas perturbações, o erro final deverá ser corrigido com o sensor de velocidade.

A resposta do sistema em laço fechado compensado em tempo discreto se mostra na Fig. 5.2.22.

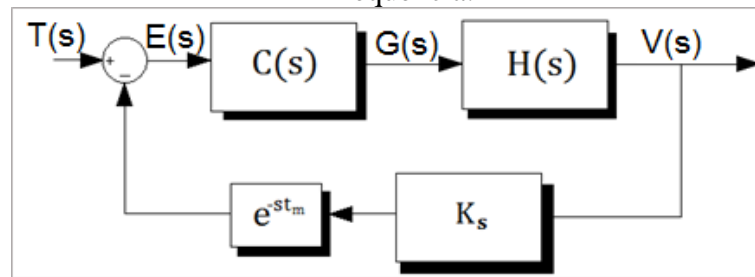
Figura 5.2.22. Resposta ao salto do sistema realimentado em tempo discreto, com um tempo de amostragem de 300 μ s.



Fonte: Próprio Autor

5.2.3.15 Estratégia de controle para o sistema em laço fechado com o atraso no sensor

Figura 5.2.23. Diagrama de blocos do motor com o feedback do sensor no domínio da frequência.



Fonte: Próprio autor.

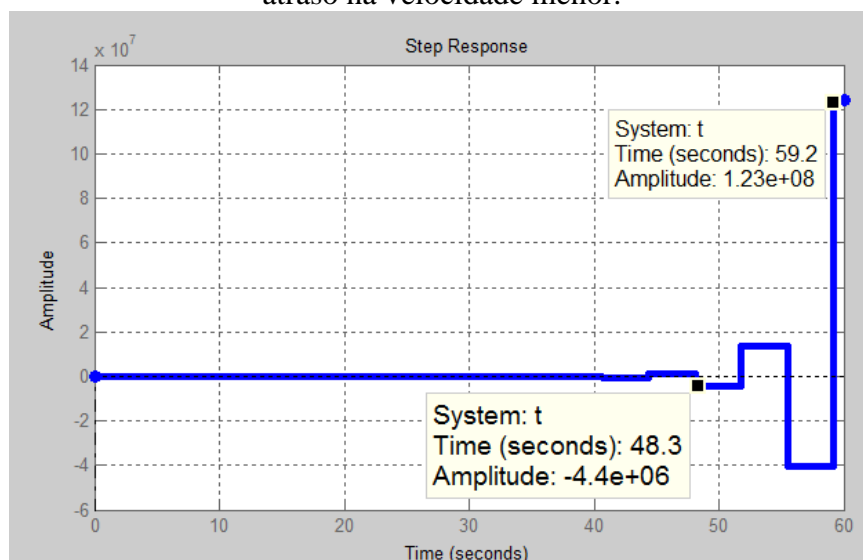
Partindo do diagrama de blocos da Fig. 5.2.23 com $K_s = 1$ e um tempo de atraso (t_m) proporcional aos valores expostos na Tab. (5.2.3), a função de transferência em laço fechado com o atraso do sensor é dada conforme à Eq. (5.2.43).

$$G_{PI}(s) = \frac{C(s) \cdot H(s)}{1 + C(s) \cdot H(s)e^{-s \cdot t_m}} \quad (5.2.43)$$

$$G(s) = \frac{288,8 \times 10^6}{s^2 + 16525 \cdot s + 388,7 \times 10^6 \cdot e^{-s \cdot t_m}}$$

Cuja resposta ao salto é apresentada na Fig. 5.2.24.

Figura 5.2.24. Resposta ao salto em laço fechado para o motor utilizando o sensor com o atraso na velocidade menor.



Fonte: Próprio autor.

A resposta ao salto em laço fechado do motor com o atraso do encoder é vista na Fig. 5.2.24, na qual se pode enxergar que o sistema responde depois dos 40 s onde entra em

instabilidade, pelo que a utilização do sensor com o nível de atraso exposto é inviável para compensação PI com o modelo encontrado na análise precedente.

Como já foi mostrado o sensor não pode ser utilizado para compensação em tempo contínuo, devido ao atraso na medição. O controle ou compensador PI calculado anteriormente, apenas será utilizado para regulagem das perturbações do sistema após esteja estabilizado usando a técnica PWM, uma vez o PWM fixe a velocidade de rotação, qualquer variação a perturbação na corrente de armadura do motor será filtrada com o modelo PI calculado, mas é o PWM que regulará a velocidade de soldagem.

Para estas circunstâncias foi escolhido o controle da velocidade através do PWM. A única variável que poderia perturbar a velocidade média do carrinho é uma variação na carga que ao longo do processo será constante.

Ao empregar o PWM para a regulagem da velocidade do motor de CC, apenas é necessário a utilização do ajuste proporcional do ciclo de trabalho do sinal PWM, a medição com o encoder permitirá o cálculo do erro entre a velocidade real e a velocidade estabelecida, podendo ser calculado o ajuste necessário para o ciclo de trabalho do PWM (as variações na velocidade final medida serão proporcionais às variações da carga), com esta modalidade o tempo de assentamento depende do atraso na medição com o encoder.

Esta é uma grande vantagem do uso da técnica PWM no controle de velocidade constante para motores de corrente contínua, a alta frequência de operação do sinal PWM depende da resposta em frequência do motor, o que estabiliza a tensão e mantém o torque estável. Para melhorar a operação do motor com a utilização desta técnica deve ser fornecida ao motor uma tensão do dobro da tensão nominal, assim o torque instantâneo do motor será do dobro do torque nominal. Dado que a tensão nominal do motor escolhido é 12 V, a tensão de pico que deverá ser fornecida para o controle por PWM deve ser 24 V, como exemplo a seguir se mostrará o que acontece com o torque do motor instantaneamente.

O torque do motor é dado pela equação a seguir (KUO, 1996, p. 177).

$$T_m = K_t I_a \quad (5.2.44)$$

A corrente instantânea do motor (sem se importar pelas variações da carga) é da forma

$$I_a = \frac{V_{24V}}{R_a} = \frac{24V}{24\Omega} = 1A \quad (5.2.45)$$

O torque mecânico produzido pelos efeitos eletromagnéticos na excitação dos bobinados na armadura instantaneamente será:

$$T_m = (0,1718 \cdot Nm/A)(1A) = 0,1718 \cdot Nm$$

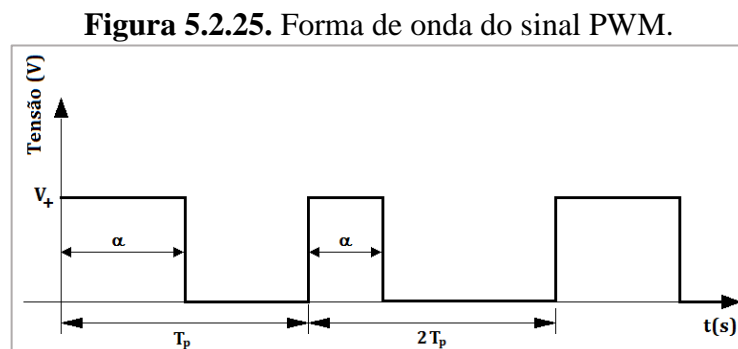
Ao usar o PWM a velocidade final do motor será proporcional à tensão média percebida pelo motor (KUO, 1996, p. 180) em um tempo muito maior a $T_{setting}$ do motor que é de $431 \mu s$.

5.2.3.16 Análise do sinal PWM no motor de CC

A série de Fourier para um sinal do tipo PWM pode ser calculada com a Eq. (5.2.46) (ALVARADO, 2008, p. 131)

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{2n\pi t}{T_p}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{2n\pi t}{T_p}\right) \quad (5.2.46)$$

O período do sinal (T_p) e o ângulo de disparo são representados na Fig. 5.2.25.



Fonte: Próprio Autor.

Do gráfico temos que $T_p = \alpha + (2\pi - \alpha)$ cálculo das constantes da série podem ser encontradas com as Eq. (5.2.47), Eq. (5.2.48) e Eq. (5.2.49), (ALVARADO, 2008, p. 134).

$$a_0 = \frac{2}{T_p} \int_0^{T_p} f(t) dt \quad (5.2.47)$$

$$a_n = \frac{2}{T_p} \int_0^{T_p} f(t) \cdot \cos\left(\frac{2n\pi t}{T_p}\right) dt \quad (5.2.48)$$

$$b_n = \frac{2}{T_p} \int_0^{T_p} f(t) \cdot \sin\left(\frac{2n\pi t}{T_p}\right) dt \quad (5.2.49)$$

O sinal do PWM da Fig. 5.2.25 pode ser representado matematicamente como

$$f(t) = \begin{cases} V_+ & 0 \leq t < \alpha \\ 0 & \alpha \leq t \leq T_p \end{cases} \quad (5.2.50)$$

Tendo que o valor máximo do ciclo de trabalho do PWM é o 50%, vai obter-se a solução para valores de $0 \leq \alpha \leq \pi$

$$a_0 = \frac{2}{2\pi} \int_0^\alpha V_+ dt = \frac{2 \cdot \alpha \cdot V_+}{2\pi} \quad (5.2.51)$$

$$a_0 = \frac{\alpha \cdot V_+}{\pi} \quad (5.2.52)$$

$$a_n = \frac{2V_+}{2\pi} \int_0^\alpha \cos\left(\frac{2n\pi t}{2\pi}\right) dt = \frac{2V_+}{\pi} \left[\sin\left(\frac{2n\pi t}{2\pi}\right) \right]_0^\alpha \quad (5.2.53)$$

$$a_n = \frac{2V_+}{2\pi} \sin\left(\frac{2 \cdot \alpha \cdot n\pi}{2\pi}\right) \quad (5.2.54)$$

$$b_n = \frac{2V_+}{2\pi} \int_0^\alpha \sin\left(\frac{2n\pi t}{2\pi}\right) dt = \frac{2V_+}{\pi} \left[-\cos\left(\frac{2n\pi t}{2\pi}\right) \right]_0^\alpha \quad (5.2.55)$$

$$b_n = \frac{2V_+}{2\pi} \left(1 - \cos\left(\frac{2 \cdot \alpha \cdot n\pi}{2\pi}\right) \right) \quad (5.2.56)$$

A série de Fourier que representa a sinal do PWM no tempo para qualquer ciclo de trabalho é dada pela expressão:

$$f(t) = \frac{\alpha \cdot V_+}{\pi} + \sum_{n=1}^{\infty} \left(\frac{2V_+}{\pi} \sin\left(\frac{2 \cdot \alpha \cdot n\pi}{2\pi}\right) \right) \cos\left(\frac{2n\pi t}{2\pi}\right) + \sum_{n=1}^{\infty} \left(\frac{2V_+}{2\pi} \left(1 - \cos\left(\frac{2 \cdot \alpha \cdot n\pi}{2\pi}\right) \right) \right) \sin\left(\frac{2n\pi t}{2\pi}\right)$$

O nosso caso particular no sistema que alimentará o motor é especificamente para valores de $0 \leq \alpha \leq \pi$ ou $0 \leq DC \leq 50\%$. Para o valor $\alpha = 0$, $f(t) = 0$, para o outro extremo $\alpha = \pi$.

$$f(t) = \frac{V_+}{2} + \sum_{n=1}^{\infty} \left(\frac{V_+}{\pi} \sin(n\pi) \right) \cos(n\pi t) + \sum_{n=1}^{\infty} \left(\frac{V_+}{\pi} (1 - \cos(n\pi)) \right) \sin(n\pi t) \quad (5.2.57)$$

Por inspeção a $f(t)$ pode ser simplificada para qualquer valor de n ímpar como:

$$f(t) = \frac{V_+}{2} + \frac{V_+}{\pi} \sum_{2n-1}^{\infty} \sin(n\pi t) \quad (5.2.58)$$

Considerando os dois primeiros componentes harmônicos

$$f(t) = \frac{V_+}{2} + \frac{V_+}{\pi} \sin(\pi t) + \frac{V_+}{\pi} \sin(3\pi t) \quad (5.2.59)$$

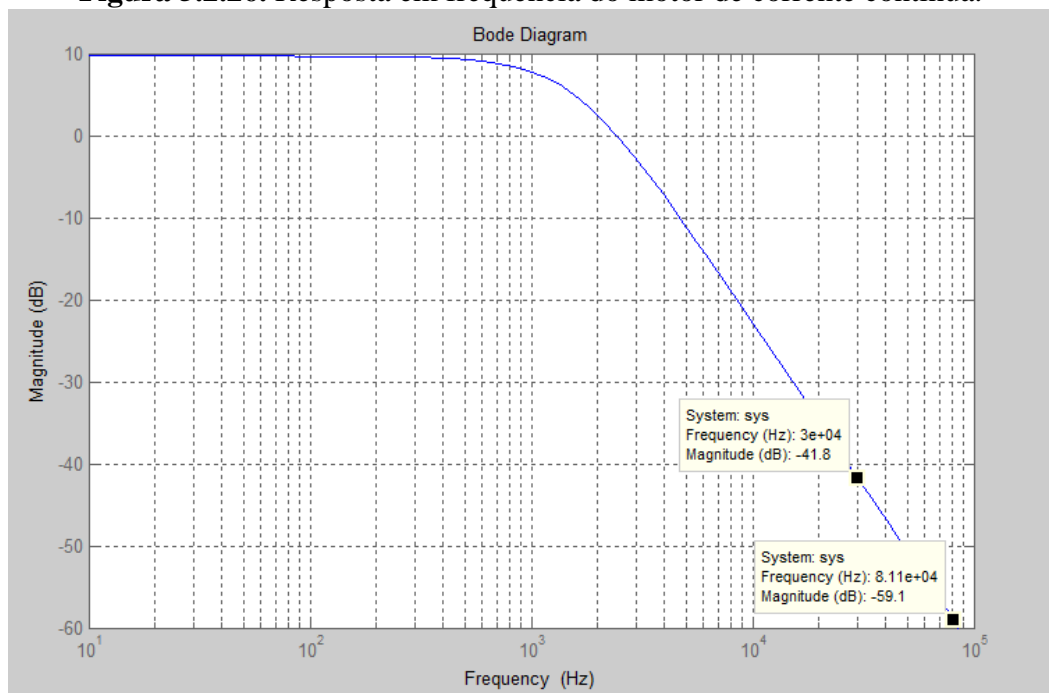
Este é o valor máximo da tensão de saída para o 50% no PWM, no circuito gerador do PWM, esta expressão em termos da frequência é dada pela função:

$$f(\omega t) = \frac{V_+}{2} + \frac{V_+}{\pi} \sin(\omega t) + \frac{V_+}{\pi} \sin(3\omega t) \quad (5.2.60)$$

A resposta em frequência do motor de CC (Eq. (5.2.18)), pode se observar na Fig. 5.2.26. Na figura se apontam os valores de atenuação para os principais componentes harmônicos do sinal PWM ao serem aplicados ao motor.

A escolha da frequência para o sinal do PWM dentro do microcontrolador e na simulação da Fig. 5.2.27, foi baseada no resultado da Eq. (5.2.21). O valor da **frequência do PWM escolhida é 30 kHz**.

Figura 5.2.26. Resposta em frequência do motor de corrente contínua.



Fonte: Próprio Autor.

O primeiro componente harmônico fundamental do PWM centrado em 30 kHz apresenta uma atenuação da sua magnitude final de -42 dB, o segundo componente harmônico abaixo de -60 dB. Utilizando uma tensão de alimentação de 24 V, se baseando na Fig. 5.2.26 e na função resultante da Eq. (5.2.60), o nível de tensão resultante visto pelo motor segundo a sua resposta em frequência pode ser representado com a Eq. (6.2.61)

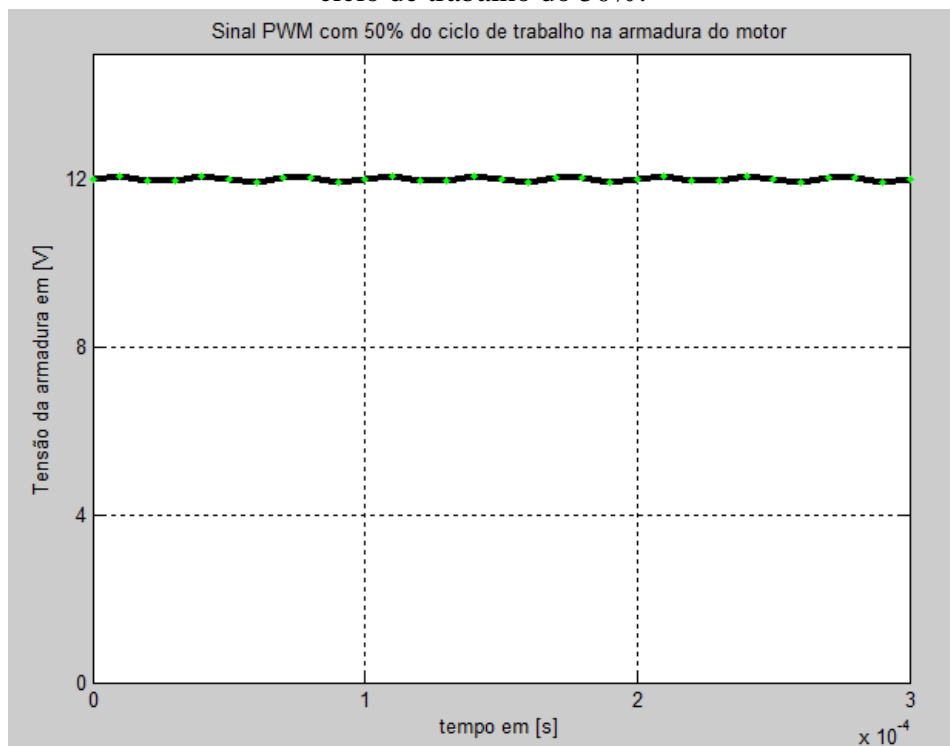
$$f(\omega t) = \frac{V_+}{2} + \frac{V_+}{125\pi} \sin(\omega t) + \frac{V_+}{1000\pi} \sin(3\omega t) \quad (5.2.61)$$

O valor de entrada visto pelo motor de CC, produto do chaveamento do sinal PWM, para essa frequência e com esse ciclo de trabalho se mostra na Eq. (5.2.62).

$$f(\omega t) = 12V + 0,06 \cdot \sin(\omega t) + 0,007 \cdot \sin(3\omega t) \quad (5.2.62)$$

A Fig. 5.2.27, foi feita para mostrar a forma da onda percebida pelo motor para um PWM com uma frequência de 30 kHz, com ciclo de trabalho de 50%, para uma fonte de 24 V.

Figura 5.2.27. Sinal PWM teórica percebida pelo motor com uma frequência de 30 kHz e um ciclo de trabalho do 50%.



Fonte: Próprio Autor.

O valor médio desse sinal é 12 V, pelo que em resumo, a tensão média da armadura percebida pelo motor para um PWM de 30 kHz e com uma fonte de 24 V, vai ser proporcional à expressão da Eq. (5.2.63).

$$V_a = \frac{24 \cdot \alpha}{\pi} \quad (5.2.63)$$

Reescrevendo essa expressão em termos do ciclo de trabalho tendo que $0 \leq \alpha \leq \pi$

$$V_a = 24 \cdot DC \quad (5.2.64)$$

$$0 \leq DC \leq 50\%$$

5.2.3.17 Projeto do gerador do sinal PWM para o motor

Para a construção do circuito gerador do PWM a partir da fonte de 24 V, é necessário um dispositivo que suporte a corrente de pico no arranque do motor de CC que é de 3,85 A. A fonte de 24 V foi escolhida por ser do dobro do valor da tensão nominal do motor que é de 12 V.

O esquema básico do PWM é apresentado na Fig. 5.2.28, o sistema digital que gera a sinal PWM para o controle é o microcontrolador PIC16F877A, a saída do PWM deste dispositivo varia entre 0 e 5 V, desta forma precisa-se de um driver de corrente chamado também de driver de potência.

Este dispositivo driver deve poder chavear numa frequência superior a 30 kHz e fornecer um ganho de 4,8 V/V.

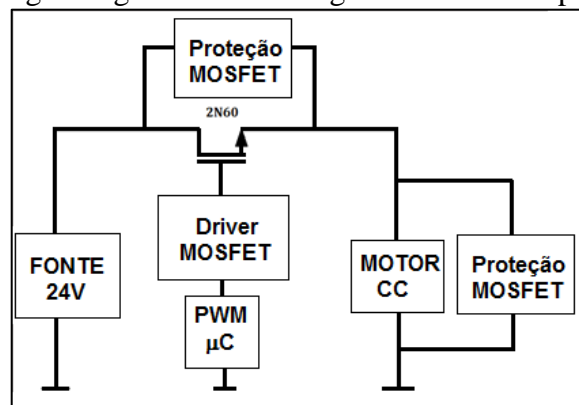
No mercado Brasileiro um dispositivo comumente utilizado para aplicações de chaveamento é o MOSFET 2N60, cujas características mais importantes são listadas na tabela a seguir.

Tabela 5.2.7. Parâmetros do MOSFET 2N60.

Parâmetro	Valor
V_{DS}	600V@150°C
V_{GS}	±30 V
$V_{GS(thr)}$	4 V
I_D	2 A
$I_D(pulsada)$	8 A
P_D	56,8W
dv/dt	5 V/μs
f_{sw}	11 MHz@2A

Fonte: DTS (2N60).

Figura 5.2.28. Diagrama geral do sistema gerador de PWM para o motor de CC.



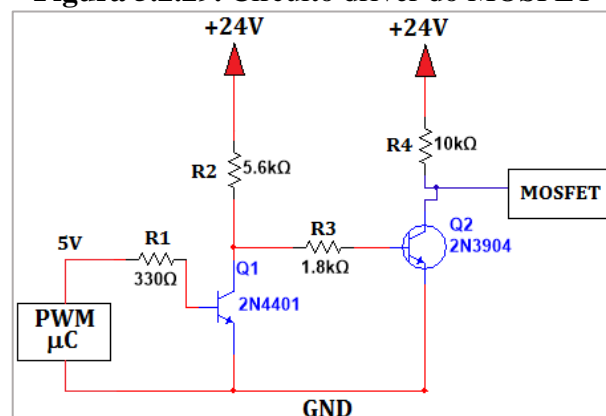
Fonte: Próprio Autor.

A Fig. 5.2.28 apresenta o diagrama de blocos do sistema gerador do sinal PWM, será necessário projetar os blocos “Driver MOSFET” e “Proteção MOSFET” e configurar o módulo PWM dentro do microcontrolador (μC) para o correto funcionamento do gerador do PWM.

Bloco Driver MOSFET

O driver MOSFET proposto é um arranjo de transistores BJT para dar o ganho da tensão necessário na saída do microcontrolador até a comporta do MOSFET. O circuito amplificador da corrente de saída do PIC16F877A apresenta-se na Fig. 5.2.29.

Figura 5.2.29. Circuito driver do MOSFET



Fonte: Próprio Autor

O cálculo dos componentes deste subcircuito foca-se nas correntes máximas e as frequências de chaveamento dos transistores. Esses parâmetros são resumidos na Tab. (5.2.8).

Tabela 5.2.8. Parâmetros dos transistores BJT

Transistor	Parâmetro	Valor
Q1 2N4401	I_C	200 mA
	V_{BE}	0,75
	V_{SAT}	0,2 V
	f_{SW}	300 MHz
Q2 2N3904	I_C	200 mA
	V_{BE}	0,85 V
	V_{SAT}	0,2 V
	f_{SW}	270 MHz

Fonte: DTS (2N3904) e (2N4401)

As correntes pelos transistores se calculam a seguir (SEDRA, 2006, p. 407).

$$I_{C(Q1)} = \frac{V_{24} - V_{SAT}}{R_1} = \frac{24V - 0,2V}{5,6k\Omega} = 4,25mA \quad (5.2.65)$$

$$I_{C(Q2)} = \frac{V_{24} - V_{SAT}}{R_4} = \frac{24V - 0,2V}{5,6k\Omega} = 2,38mA \quad (5.2.66)$$

As potências dos resistores são as seguintes (ALEXANDER e SADIKU, 2006, p. 33).

$$P_{R1} = I_{R1} * V_{R1} = 53\mu A * (53\mu A * 330\Omega) = 0,9\mu W \quad (5.2.67)$$

$$P_{R2} = I_{R2} * V_{R2} = 4,25mA * (24V - 0,2V) = 101mW \quad (5.2.68)$$

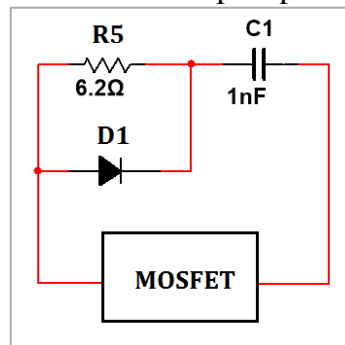
$$P_{R3} = I_{R3} * V_{R3} = 3,12mA * \left(\frac{1,8 * (24V - 0,85V)}{7,4} \right) = 17mW \quad (5.2.69)$$

$$P_{R4} = I_{R4} * V_{R4} = 2,38mA * (24V - 0,2V) = 56mW \quad (5.2.70)$$

Blocos de proteção MOSFET

Para a proteção do MOSFET se integrará uma rede SNUBBER RDC (Resistor-Diodo-Capacitor) contra sobre tensões em paralelo com o MOSFET (RASHID, 1995, p. 597).

Figura 5.2.30. Rede SNUBBER para proteção do MOSFET.



Fonte: Modificado de Rashid (1995, p. 597).

Para o cálculo da rede SNUBBER é necessário saber a constante dv/dt do MOSFET que para o 2N60 é de 5 V/ns. Esse parâmetro pode ser inserido na equação da corrente para a capacitância Drenador-Surtidor (Drain-Source) (PRESMAN, 2009, p. 465).

$$i_M = C_1 \frac{dv}{dt} \quad (5.2.71)$$

A tensão instantânea no capacitor em função da sua tensão média é dada pela Eq. (5.2.72) (ALEXANDER e SADIKU, 2006, p. 275)

$$V_C = V_f(1 - e^{t/R_5 C_1}) \quad (5.2.72)$$

Sendo $V_f = I_m R_5$, onde I_m é a corrente máxima da carga (o motor).

$$V_C = I_m R(1 - e^{-t_{ON}/R_5 C_1}) \quad (5.2.73)$$

Na última equação têm-se duas condições

$$V_C = I_m R_5 \quad (5.2.74)$$

$$0 \approx e^{-\frac{t_{ON}}{RC}} \quad (5.2.75)$$

Para que a primeira condição seja satisfeita precisa-se que na segunda condição.

$$t_{ON} \geq 5RC \quad (5.2.76)$$

Ou seja:

$$R \leq \frac{t_{ON}}{5C} \quad (5.2.77)$$

Sem esquecer que $R_5 = V_C/I_m$. Se as duas condições são satisfeitas então $V_C = V_S$, onde V_S é a tensão da fonte. Na última equação t_{ON} é o tempo mínimo do sinal PWM.

Finalmente o valor do capacitor na rede SNUBBER pode ser encontrado com

$$C = \frac{\left(\frac{dv}{dt}\right)}{i_M} \quad (5.2.78)$$

Para o caso do sistema atual os valores calculados são:

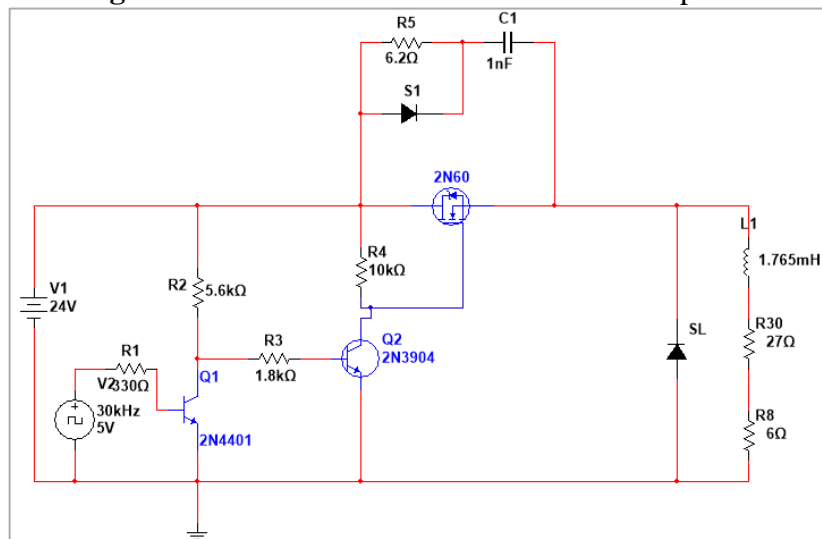
$$C_1 = \frac{\left(\frac{5V}{ns}\right)}{3.8A} = \frac{5}{3.8} \times 10^{-9} \approx 1nF$$

$$R_5 = \frac{24V}{3.8A} \approx 6.3\Omega$$

$$R_5 \leq \frac{t_{ON}}{5C} = \frac{1\% T_{PWM}}{5C} = 66.6 \Omega$$

Onde $T_{PWM} = 33.3 \mu s$ e o 1% é 333.3 ns, o diodo S1 escolhido foi o HER507. Finalmente o circuito de proteção do lado do motor, será o diodo SL, que é um dispositivo de rodada livre de recuperação rápida similar ao HER507 com capacidade para 4 A e tempo de recuperação de 250 ns.

Figura 5.2.30. Sistema Gerador do PWM completo.



Fonte: próprio autor.

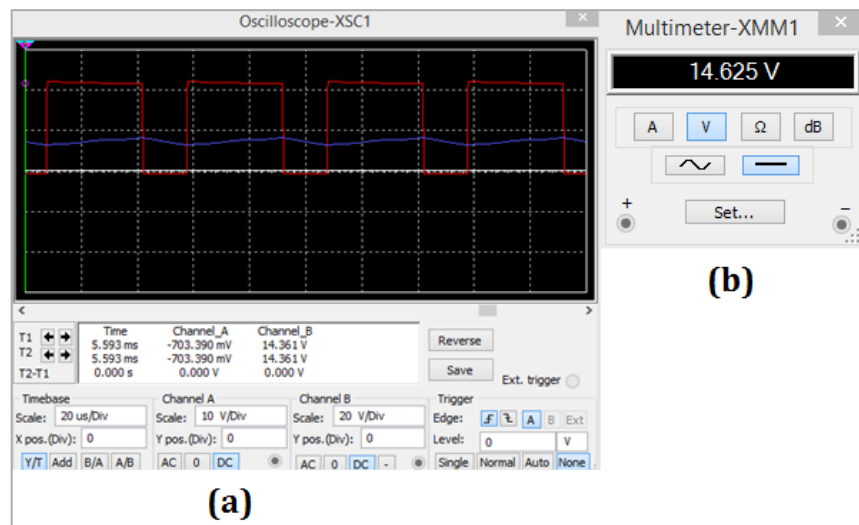
Simulação do circuito gerador do sinal PWM

O sinal PWM que abre e fecha o MOSFET é fornecida pelo microcontrolador, que na simulação é substituído por uma fonte ideal com uma frequência de 30 kHz (frequência de

chaveamento). O indutor L1 e o resistor R30 vistos na Fig. 5.2.30, representam o motor de corrente contínua, a resistência R8 é a resistência empregada para medir a corrente do motor dentro do microcontrolador através da porta do conversor analógico digital (ADC).

Na Fig. 5.2.31 mostra-se a simulação do circuito da Fig. 5.2.30, para um ciclo de trabalho de 68%, cuja tensão de saída é 14.625 V contínuos que é o limite máximo de tensão permissível pelo motor.

Figura 5.2.31. Simulação do sistema gerador PWM com um ciclo de trabalho de 68%, (a) Sinal no osciloscópio que entra no motor, PWM (vermelho), I_m (azul); (b) Tensão média do motor.



Fonte: próprio Autor.

Para o caso de uma tensão de saída no gerador de PWM de 14.625 V a velocidade de giro do motor será de 12,68 rpm, pela relação das polias entre o motor e a roda do carrinho (2:1) a velocidade das rodas será de 6,34 rpm, agora com ajuda da Eq. (5.2.79) pode ser calculada a velocidade da soldagem.

$$V_{el} = 2\pi r \cdot RPM \left[\frac{cm}{min} \right] \quad (5.2.79)$$

A velocidade de soldagem para essa rotação do motor é calculada a seguir.

$$V_{el} = 2\pi(3cm) \cdot (6.34 rpm) \left[\frac{cm}{min} \right] = 119.5 \frac{cm}{min}$$

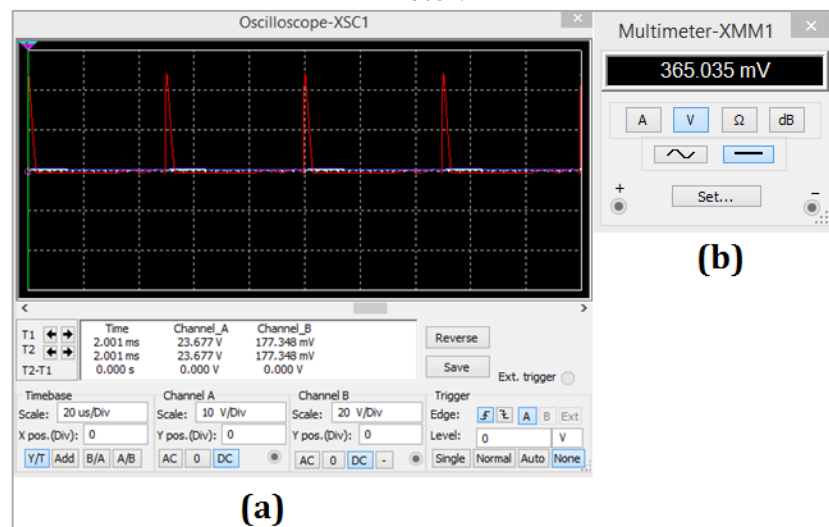
Esse valor do ciclo de trabalho é o máximo que deveria se impor no gerador de PWM, um ciclo de trabalho além de 68% danificaria o motor.

Teoricamente para uma tensão de saída de 365 mV a velocidade de giro do motor é de 0,32 rpm, o giro do motor é dividido por dois devido a relação das polias então a velocidade de soldagem para esta tensão é dada a seguir.

$$V_{el} = 2\pi(3 \text{ cm}) \cdot (0,16 \text{ rpm}) = 3 \left[\frac{\text{cm}}{\text{min}} \right] \quad (5.2.80)$$

Esse valor de tensão é obtido na simulação da tensão de saída do gerador de PWM e é apresentado na Fig. 5.2.32, para um ciclo de trabalho de 1%. Esses valores de velocidade foram calculados considerando um raio das rodas do robô igual a 3 cm.

Figura 5.2.32. Simulação do sistema gerador PWM com um ciclo de trabalho do 1%, (a) Sinal no osciloscópio que entra no motor, PWM (vermelho), I_m (azul); (b) Tensão média do motor.



Fonte: próprio Autor.

O nível de programação no algoritmo de controle as velocidades estabelecidas foram de entre 16 – 110 cm/min. A principais razões são:

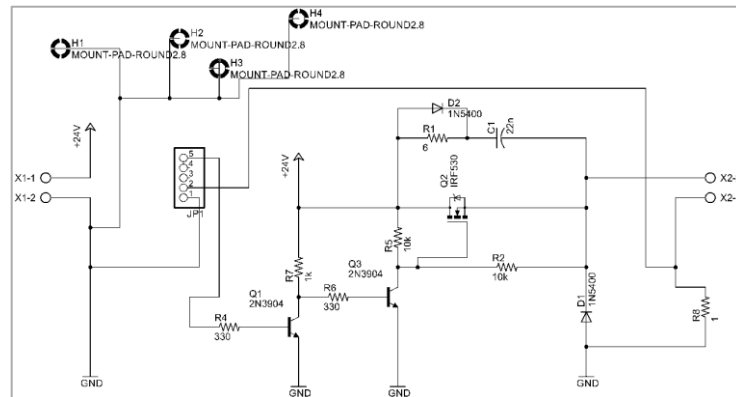
1. A tensão de rompimento do motor é de 1,35 V, abaixo desse valor o comportamento de velocidade não é linear, 16 cm/min equivale ao 10% do ciclo de trabalho ou 1,5 V aproximadamente.
2. Diminuir o tempo de estabilização do sistema a menos de 4 s na velocidade menor.

5.3.2.18 Circuito projetado para regulação do PWM

O esquemático do circuito montado no software Cadsoft Eagle, é apresentado na Fig. 5.2.33, o componente 2N60 foi substituído pelo transistor IRF530 que possui a mesma

configuração de pinos, mas esta substituição foi simplesmente para geração do layout e implementação do circuito impresso.

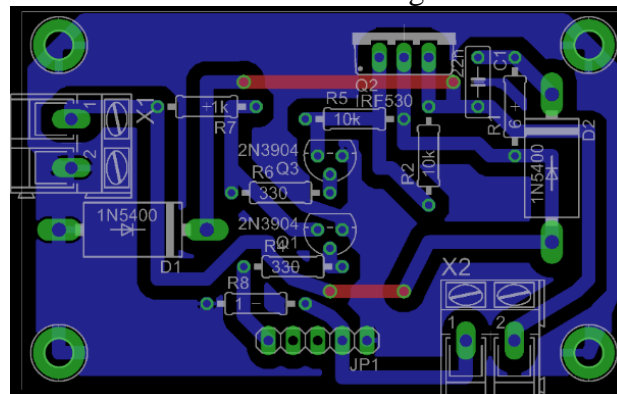
Figura 5.2.33. Circuito regulador PWM com MOSFET 2N60 montado no Cadsoft Eagle V5.0.



Fonte: Próprio Autor.

O circuito impresso criado e gerado com o software Eagle é apresentado na Fig. 5.2.34, a geração das pistas foi feita manualmente para garantir uma ligação apropriada e compatível com a aplicabilidade na frequência do PWM estabelecida.

Figura 5.2.34. Layout do circuito impresso para o regulador PWM com o MOSFET 2N60 montado no Cadsoft Eagle V5.0.

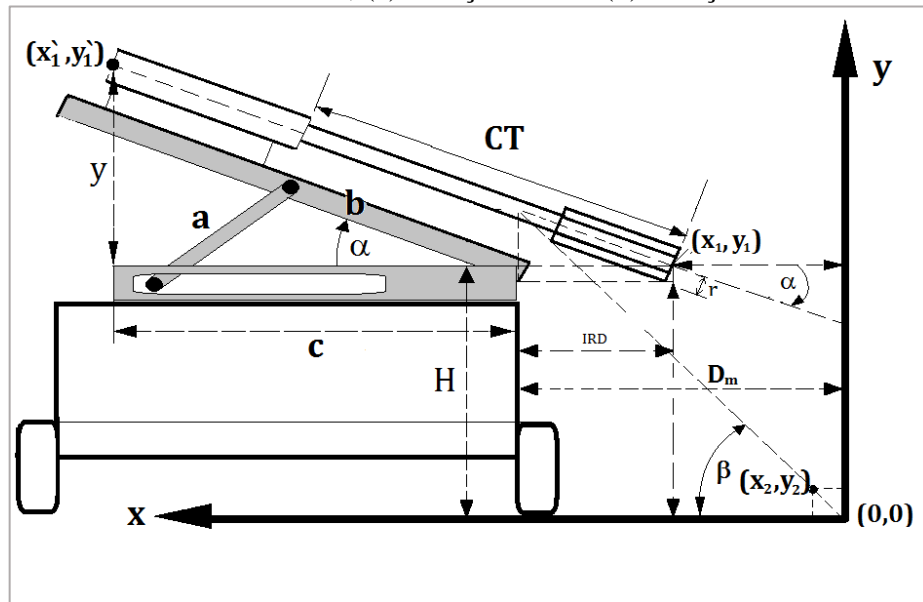


Fonte: Próprio Autor.

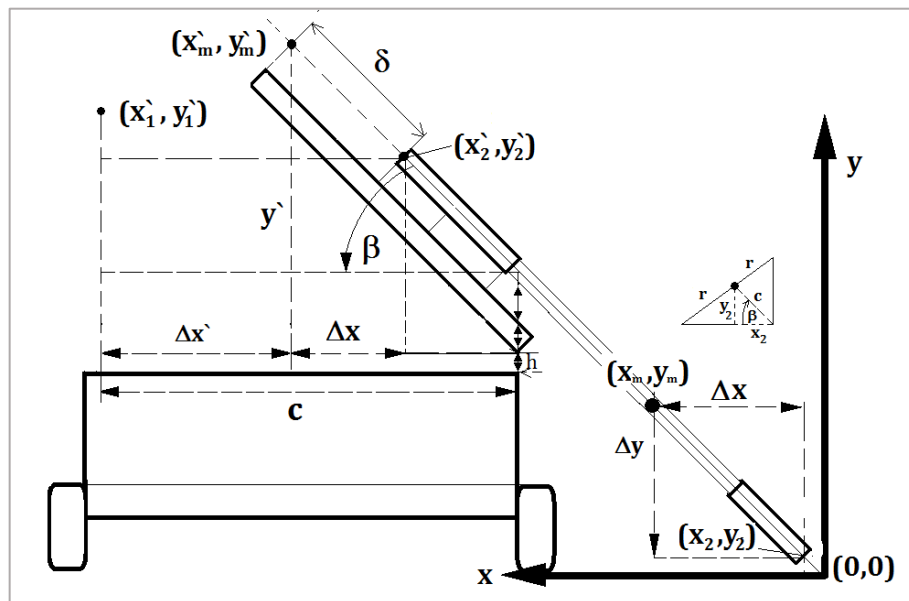
5.2.4 Bloco Sistema de posicionamento CNC

5.2.4.1 Obtenção das equações geométricas de posicionamento

Figura 5.2.35. Modelagem geométrica do sistema angular para a obtenção das equações de controle numérico, (a) Posição inicial (b) Posição final.



(a)



(b)

Fonte: Próprio Autor

Todas as equações obtidas a seguir foram extraídas das Fig. 5.2.35(a) e Fig. 5.2.35(b), essas equações descrevem os deslocamentos do sistema de posicionamento. Durante o posicionamento da tocha o bico de contato na inclinação do mecanismo traçará uma trajetória circular cujo ponto máximo (x_c, y_c) é descrito pelas Eq. (5.2.81) e Eq. (5.2.82).

$$x_c = CT \cdot \cos(\alpha) + A \cdot \sin(\alpha) - 62,13 \quad (5.2.81)$$

$$y_c = A \cdot \cos(\alpha) - CT \cdot \sin(\alpha) + r \cdot \frac{\cos(\alpha)}{\sin(\alpha)} \quad (5.2.82)$$

A distância mínima de colocação do carrinho afastado da peça vertical para a soldagem é dada pela Eq. (5.2.83).

$$D_m(\min) = \sqrt{(x_c)^2 + (y_c)^2} \pm 1mm \quad (5.2.83)$$

Para obter algumas expressões é necessário definir algumas variáveis auxiliares, duas delas são expostas na Eq. (5.2.84) e na Eq. (5.2.85).

$$dsm = 62,13 \cdot \cos(\alpha) \quad (5.2.84)$$

$$x = CT \cdot \cos(\alpha) \quad (5.2.85)$$

Os parâmetros do sistema mecânico a serem medidos e calculados partindo do esquemático da Fig. 5.2.35 apresentam-se a seguir.

$$I_{RD} = x - dsm = (CT - 62,13) \cdot \cos(\alpha) \quad (5.2.86)$$

$$I_{RD}' = (CT - 62,13) \cdot \cos(\beta - \alpha) \quad (5.2.87)$$

$$x_m = D_m - I_{RD}' \quad (5.2.88)$$

$$\Delta x = x_m - x_2 \quad (5.2.89)$$

$$x_2 = r \cdot \frac{\cos(\beta)^2}{\sin(\beta)} \quad (5.2.90)$$

$$\delta = \frac{\Delta x}{\cos(\beta - \alpha)} \quad (5.2.91)$$

Desenvolvendo a Eq. (5.2.91) obtém-se a expressão necessária para o deslocamento da mesa linear, no desenvolvimento são utilizadas as Eq. (5.2.87), Eq.(5.2.88), Eq. (5.2.89) e Eq. (5.2.90).

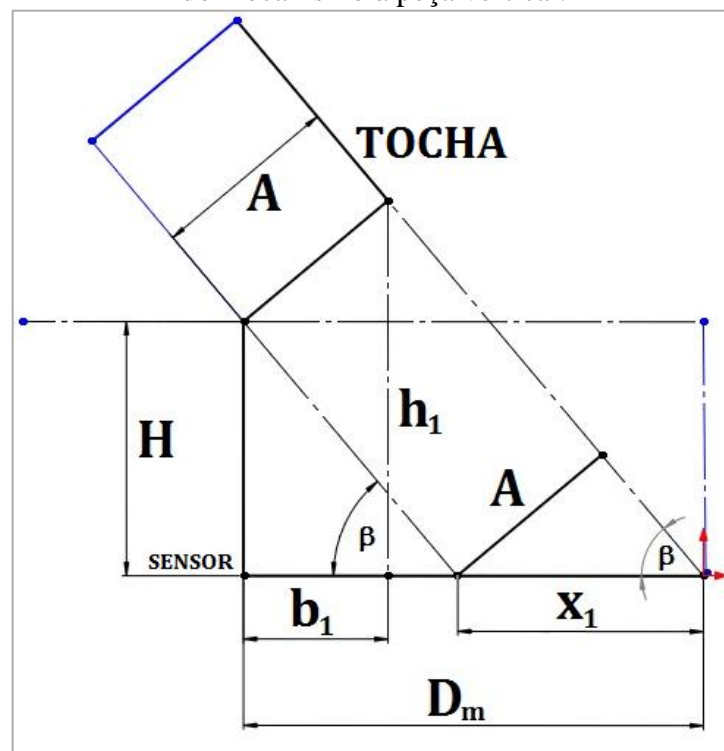
$$\delta = \frac{(x_m - x_2)}{\cos(\beta - \alpha)}$$

$$\delta = \frac{D_m - I_{RD}' - r \cdot \frac{\cos(\beta)^2}{\sin(\beta)}}{\cos(\beta - \alpha)}$$

$$\delta = \frac{D_m - (CT - 62,13)\cos(\beta - \alpha) - r \frac{\cos^2(\beta)}{\sin(\beta)}}{\cos(\beta - \alpha)} \quad (5.2.92)$$

A Eq. (5.2.92) representa o deslocamento linear necessário da mesa que suporta a tocha para atingir a peça de soldagem. O cálculo do número de passos do motor utilizado neste mecanismo, será efetuado baseado na Eq. (5.2.92) e um tamanho de passo do fuso de 1 mm por volta.

Figura 5.2.36. Geometria do robô para obtenção do ângulo de soldagem baseado na distância do mecanismo à peça vertical.



Fonte: Próprio autor.

Da Fig. 5.2.36 precedente extraíram-se as equações a seguir, produto das constantes do mecanismo.

$$h_1 = H + A \cdot \cos(\beta) \quad (5.2.93)$$

$$b_1 = A \cdot \sin(\beta) \quad (5.2.94)$$

$$A = D_m \cdot \sin(\beta) - H \cdot \cos(\beta) \quad (5.2.95)$$

$$\tan(\beta) = \frac{h_1}{D_m - b_1} \quad (5.2.96)$$

Das equações precedentes podem-se obter as relações necessárias para o controle da posição. Substituindo as Eq. (5.2.93) e Eq. (5.2.94) na Eq. (5.2.96) obtém-se a expressão a seguir.

$$\tan(\beta) = \frac{H + A \cdot \cos(\beta)}{D_m - A \cdot \sin(\beta)} \quad (5.2.97)$$

Para resolver a expressão anterior pode-se isolar o seno da Eq. (5.2.92)

$$\sin(\beta) = \frac{A + H \cdot \cos(\beta)}{D_m} \quad (5.2.98)$$

Substituindo a Eq. (5.2.98) na (5.2.97) obtém-se uma expressão em termos do cosseno do ângulo correto para a soldagem baseado na distância de colocação do robô à peça vertical.

$$\frac{A + H \cdot \cos(\beta)}{D_m} = \frac{(H + A \cdot \cos(\beta))D_m \cos(\beta)}{D_m^2 - (A^2 + A \cdot H \cdot \cos(\beta))} \quad (5.2.99)$$

Resolvendo a Eq. (5.2.99) em termos do cosseno, isto para gerar uma expressão programável no microcontrolador para o controle numérico.

$$\cos^2(\beta) + \frac{2 \cdot A \cdot H}{H^2 + D_m^2} \cdot \cos(\beta) - \left(\frac{D_m^2 - A^2}{H^2 + D_m^2} \right) = 0 \quad (5.2.100)$$

Fazendo o câmbio de variável $x = \cos(\beta)$

$$x^2 + \left(\frac{2 \cdot A \cdot H}{H^2 + D_m^2} \right) \cdot x - \left(\frac{D_m^2 - A^2}{H^2 + D_m^2} \right) = 0 \quad (5.2.101)$$

A solução da Eq. (5.2.101) é da forma:

$$x_1 = \frac{\sqrt{\Delta Det} - b_0}{2a_0} \quad (5.2.102)$$

Tendo que:

$$a_0 = 1 \quad b_0 = \left(\frac{2 \cdot A \cdot H}{H^2 + D_m^2} \right) \quad c_0 = - \left(\frac{D_m^2 - A^2}{H^2 + D_m^2} \right) \quad \Delta Det = b_0^2 - 4a_0c_0$$

$$\Delta Det = \left(\frac{2 \cdot A \cdot H}{H^2 + D_m^2} \right)^2 + 4 \left(\frac{D_m^2 - A^2}{H^2 + D_m^2} \right) \quad (5.2.103)$$

Voltando da substituição da variável para a variável original

$$\cos(\beta) = \frac{\sqrt{\Delta Det} - b_0}{2a_0} \quad (5.2.104)$$

No caso explícito foram medidas as constantes do sistema envolvidas na Eq. (5.2.103) as quais se alistam na tabela a seguir

Tabela 5.2.8. Constantes medidas no robô para o controle do ângulo de soldagem no posicionamento da tocha.

Símbolo	Valor (mm)
H	107,00
A	83,50

Fonte: Próprio autor.

Os resultados de utilizar os valores da Tab. (5.2.8), nas Eq. (5.2.103) e Eq. (5.2.104) mostram-se a seguir.

$$\Delta Det = \left(\frac{17800}{11450 + D_m^2} \right)^2 + 4 \left(\frac{D_m^2 - 6972}{11450 + D_m^2} \right)$$

$$b_0 = \frac{17800}{11450 + D_m^2}$$

$$\cos(\beta) = \frac{\sqrt{\left(\frac{17800}{11450 + D_m^2} \right)^2 + 4 \left(\frac{D_m^2 - 6972}{11450 + D_m^2} \right)} - \left(\frac{17800}{11450 + D_m^2} \right)}{2} \quad (5.2.105)$$

$$\text{sen}(\beta) = \sqrt{1 - \cos(\beta)^2} \quad (5.2.106)$$

Utilizando a lei dos cossenos pode encontrar uma expressão para a estrutura trigonométrica vista na Fig. 5.2.25(a) que permita deslizar a barra (a) do mecanismo para transformar o ângulo inicial α , no ângulo requerido β .

$$a^2 = b^2 + c^2 - 2bc \cdot \cos(\beta) \quad (5.2.107)$$

$$c'^2 - 2b \cdot \cos(\beta) \cdot c' + b^2 - a^2 = 0 \quad (5.2.108)$$

Da Fig. 5.2.25(b), o novo valor da variável (c) agora será da forma $c' = c - \Delta x'$

$$\Delta x'^2 - \Delta x' 2(c - b \cdot \cos(\beta)) - (2b \cdot \cos(\beta)c - c^2 + a^2 - b^2) = 0$$

$$\Delta x' = c - b \cdot \cos(\beta) - \frac{\sqrt{4(c - b \cdot \cos(\beta))^2 + 4(2b \cdot c \cdot \cos(\beta) - c^2 + a^2 - b^2)}}{2} \quad (5.2.109)$$

No entanto o valor de (c) na Eq. (5.2.109) depende do ângulo α (ângulo inicial do mecanismo angular visto na Fig. 5.2.35 (a)). O valor máximo medido desta constante foi de aproximadamente 118 mm , então pela lei dos cossenos.

$$c^2 - 2b \cdot \cos(\alpha) \cdot c + b^2 - a^2 = 0 \quad (5.2.110)$$

$$c = b \cdot \cos(\alpha) - \sqrt{b^2 \cdot \cos^2(\alpha) - (b^2 - a^2)} \quad (5.2.111)$$

$$c = b \cdot \cos(\alpha) + \sqrt{a^2 - b^2 \cdot \sin^2(\alpha)} \quad (5.2.112)$$

As expressões na Eq. (5.2.109) e na Eq. (5.2.112) poderão ser utilizadas como funções genéricas para posicionamento angular, para qualquer valor de α , no caso for incluído um sensor de ângulo no sistema. Atualmente pelas folgas mecânicas é preciso saber o ângulo inicial para que o posicionamento seja efetuado.

A Eq. (5.2.109) é a variável de deslocamento linear necessária para variar o ângulo da tocha baseado na medição da distância do sensor à peça a se soldar. O cálculo do número de passos do motor será encontrado a partir desta variável sabendo que com cada volta do motor de passo o eixo linear deslocara-se 5 mm .

Os valores das constantes medidas no sistema mecânico estão dados na Tab. (5.2.9).

Tabela 5.2.9. Constantes medidas no robô e parâmetros gerais do sistema angular.

Símbolo	Valor (mm)	Parâmetro	Valor (mm)	Símbolo	Valor (mm)
a	68	α	21°	CT	190
b	57	r	11	x	168
c	118	dsm	177	$D_m(\text{min})$	147

Fonte: Próprio autor.

Substituindo as constantes da Tab. (5.2.9) na Eq. (5.2.92), na Eq. (5.2.109) e na Eq. (5.2.112), obtêm-se as funções para o controle numérico dos deslocamentos lineares para o posicionamento da tocha de soldagem.

$$\Delta x' = 118 - 57 \cdot \cos(\beta) - \sqrt{(118 - 57 \cdot \cos(\beta))^2 + 13450 \cdot \cos(\beta) - 12550} \quad (5.2.113)$$

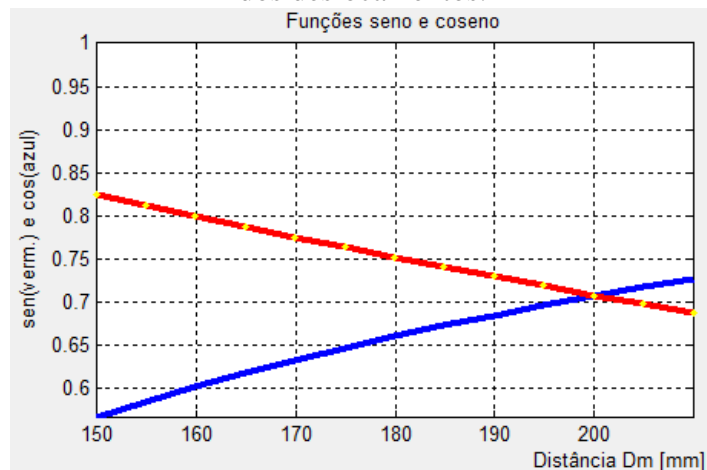
$$\delta = \frac{D_m - 119 \cdot \cos(\beta) - 46 \cdot \sin(\beta) - 11 \frac{\cos^2(\beta)}{\sin(\beta)}}{0,93 \cdot \cos(\beta) + 0,36 \cdot \sin(\beta)} \quad (5.2.114)$$

Os resultados desses deslocamentos lineares serão transformados para quantidade de passos necessários que deverão acontecer nos motores de passo segundo o avanço de cada fuso em cada mecanismo.

Por exemplo, para o fuso angular utilizado o avanço linear por volta é de 5 mm , se a função ($\Delta x'$) der um deslocamento resultante de 30 mm o motor de passo deverá dar 6 voltas, para o caso de um motor convencional de 200 passos por volta, deverão se gerar 1200 passos para atingir essa quantidade de voltas e conseguir um avanço linear de 30 mm .

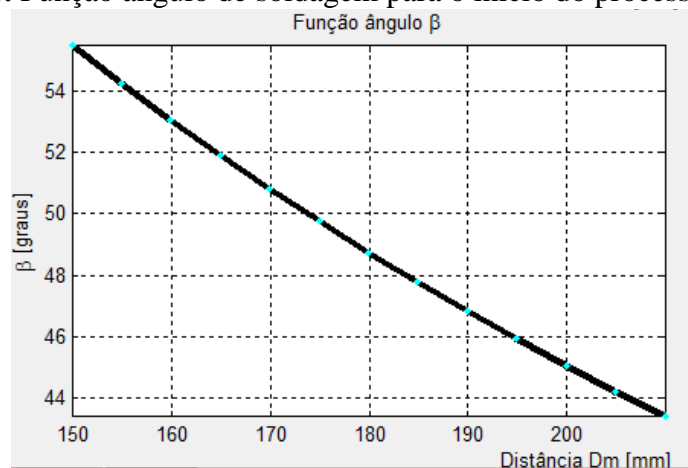
A seguir são mostrados os gráficos das Eq. (5.2.105), Eq. (5.2.106), Eq. (5.2.113) e Eq. (5.2.114) a implementar no sistema digital, nas Fig. 5.2.37, Fig. 5.2.38, Fig. 5.2.39 e Fig. 5.2.40, em função da distância do carrinho às peças (D_m).

Figura 5.2.37. Função seno e cosseno do sistema angular para o cálculo das aproximações dos deslocamentos.



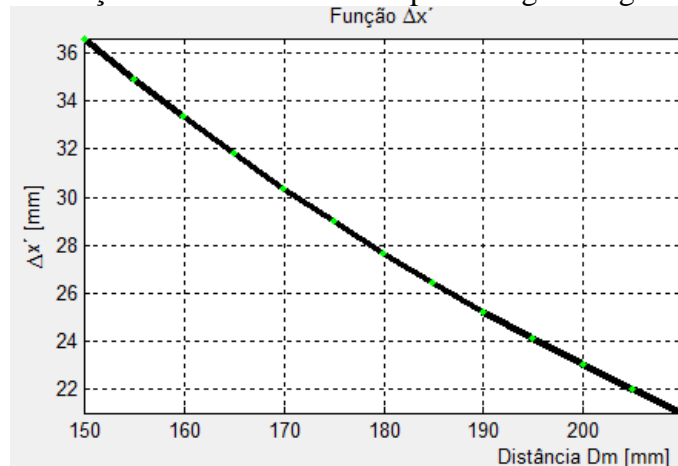
Fonte: Próprio autor.

Figura 5.2.38. Função ângulo de soldagem para o início do processo de soldagem.



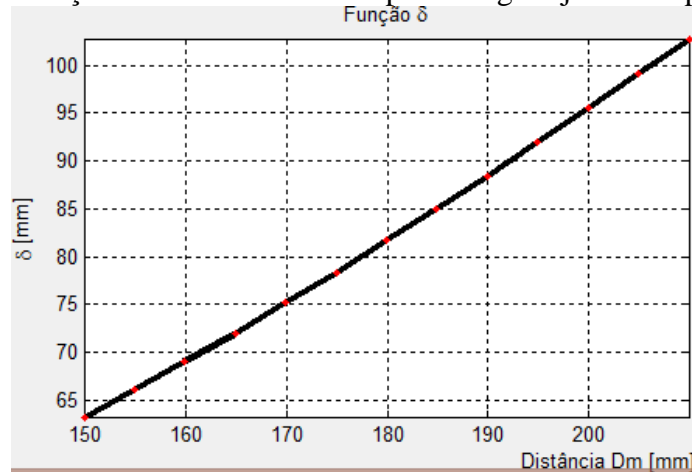
Fonte: Próprio Autor.

Figura 5.2.39. Função deslocamento linear para atingir o ângulo de soldagem.



Fonte: Próprio Autor.

Figura 5.2.40. Função deslocamento linear para atingir a junta das peças metálicas.



Fonte: Próprio Autor.

5.2.5 Projeto do sistema de controle dos motores de passo para o sistema linear e o sistema angular

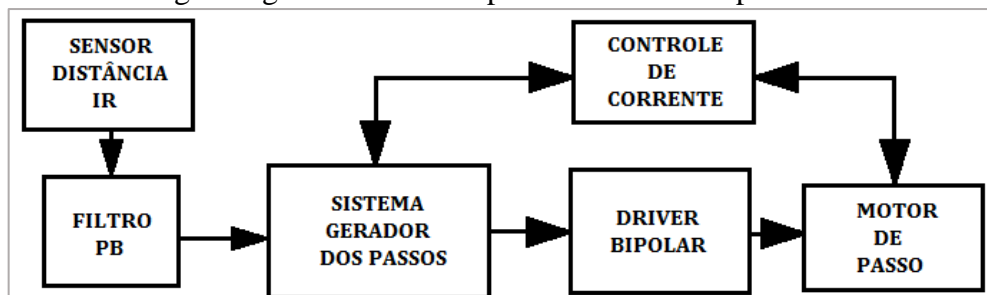
Os motores de passo escolhidos para a operação do sistema angular são motores de passo unipolares os quais são os mais econômicos, contudo esses motores podem funcionar em modo bipolar arranjando as ligações elétricas, e com isto aumentando a potência mecânica.

Um motor de passo unipolar em modo bipolar fornece o dobro do torque na velocidade de operação do unipolar que é uma velocidade média. A velocidade de operação de um motor de passo unipolar geralmente é de uma revolução por segundo.

A escolha desta característica principalmente é para compensar as incertezas de fabricação do mecanismo no processo de construção, dado que a maioria das peças foi feita em equipamentos com baixa precisão, então o erro no cálculo das forças não poderia ser medido com instrumentação convencional e as aproximações matemáticas apenas servem de orientação no projeto do sistema.

Dito todo o anterior o diagrama do controle proposto para os motores de passo se mostra na Fig. 5.2.41, cada um dos blocos dessa figura será explicado nas secções a seguir.

Figura 5.2.41. Diagrama geral do controle para os motores de passo no sistema angular.



Fonte: Próprio Autor.

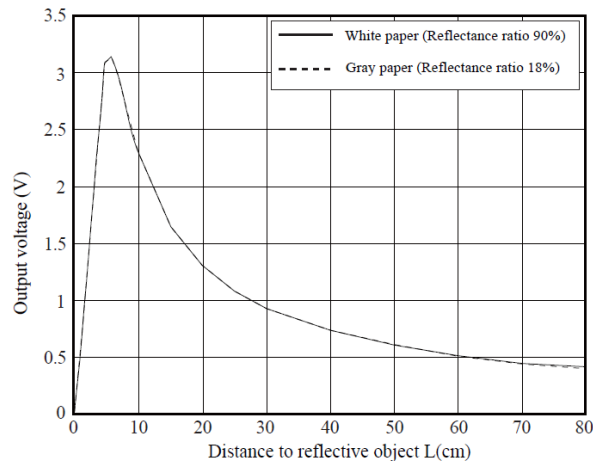
5.2.5.1 Bloco sensor de distância IR

Optou-se por um dispositivo eletrônico de detecção por luz infravermelha para sensoriamento da distância no processo de posicionamento da tocha, este dispositivo não pode interferir aos dois sensores ultrassônicos utilizados no sistema de correção da trajetória.

O carrinho vai utilizar sensores de distância por ultrassom para medir e corrigir a trajetória ao longo do processo, utilizar outro sensor ultrassônico iria interferir neste processo de detecção e controle. Para evitar qualquer interferência o sensor para o posicionamento da tocha vai ser através de luz infravermelha.

O sensor de distância por infravermelho pelas suas características no receptor é um dispositivo não linear e a faixa de operação é limitada. A Fig. 5.2.42 mostra a curva característica de funcionamento fornecida pelo fabricante para o dispositivo de medição de distância o GP2Y0A21YK da SHARP.

Figura 5.2.42. Características de saída do sensor GP2Y0A21YK da SHARP.



Fonte: DTS (GP2Y0A21YK) da SHARP.

Como pode se enxergar na Fig. 5.2.42, o sensor pode ser utilizado para medir distâncias entre 7 e 80 cm (aproximadamente), dados a serem considerados no projeto do sistema, já que a distância menor do carrinho à peça deve ser estabelecida no projeto final, para evitar erros de leitura e equívocos nos cálculos dos passos para o motor.

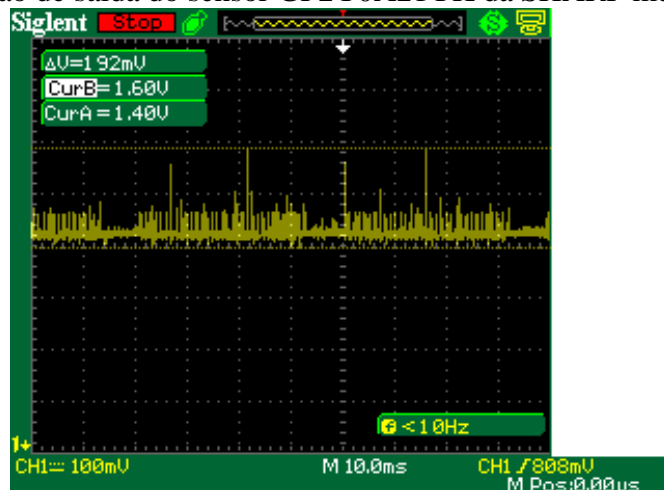
Baseado nesta característica de distância mínima a se medir pelo sensor GP2Y0A21YK foi estabelecida a distância mínima de colocação do carrinho para iniciar o processo de posicionamento da tocha, foi uma decisão estratégica pelas limitações do sistema de medição.

Esta distância será definida em 150 cm e foi calculada com ajuda da Eq. (5.2.83) que também depende do comprimento da tocha de soldagem e do ângulo inicial do sistema angular.

5.2.5.2 Bloco Filtro PB

A Fig. 5.2.42 é um gráfico feita pelo fabricante como orientação para o projetista, no entanto foi necessário caracterizar o sensor para determinar o tipo de filtragem necessária para estabilizar os dados na saída do sensor. O sinal de saída do sensor para uma distância qualquer pode ser observado na Fig. 5.2.43.

Figura 5.2.43. Tensão de saída do sensor GP2Y0A21YK da SHARP medida no osciloscópio.



Fonte: Próprio Autor.

Para o correto processamento da tensão de saída dentro do sistema gerador de passos é necessário programar dentro do microcontrolador um filtro passa baixa para evitar esses picos de ruído elétricos vistos na Fig. 5.2.43 interfiram no valor real de distância a se medir. Não se entrará em grande detalhe no projeto do filtro ao não ser um objetivo do trabalho, mas cabe ressaltar que a sintonização deste filtro consumiu grande parte do tempo no desenvolvimento do sistema de controle angular, um erro na medição desta distância faria que o sistema por completo não funcionasse. O código da programação do bloco Filtro PB dentro do microcontrolador se mostra na Fig. 5.2.44.

Figura 5.2.44. Código de programação em linguagem C correspondente ao bloco Filtro PB para o processamento do sinal fornecida pelo sensor GP2Y0A21YK da SHARP.

```

//////////////////////////////////FILTRO DO SENSOR DE DISTANCIA IR

if(!DISABLE_SENSOR){
set_adc_channel(0);
AUX0 = FIRO;
FIRO = 50000/read_adc() + FIRO/2;
if((FIRO<(105*AUX0/100))&&(VMAX<FIRO)) VMAX = FIRO;
if((FIRO>(95*AUX0/100))&&(VMIN>FIRO)) VMIN = FIRO;
if(i < 2047)i = i + 1;
else{
i = 0;
FIRO0 = (VMAX + VMIN)/2;
VMAX = FIRO0;
VMIN = FIRO0;
}
}
//////////////////////////////////FIN DEL FILTRO DE DISTANCIA

```

Fonte: Próprio autor.

O microcontrolador faz a aquisição dos dados do sensor através da porta analógica (no Conversor Analógico-Digital, ADC), a frequência de amostragem do ADC é de uma amostra cada $2 \mu\text{s}$ (500000 amostras por segundo ou 500 ksp/s), no entanto dentro do microcontrolador existem outros processos que são executados antes e depois da aquisição, isto faz que a frequência real de amostragem sem filtragem seja de 50 ksp/s (10 vezes menor pelos processos dentro do microcontrolador, medida de forma aproximada).

O código da Fig. 5.2.44 adquire 2047 amostras e faz a média simples desses dados reduzindo a frequência do filtro a 24 Hz (aproximadamente), tirando da medição o ruído elétrico do ambiente gerado pelas lâmpadas e pela rede de alimentação de 60 Hz. Além disso, foi programado um filtro discriminador que descarta durante a aquisição valores acima do $\pm 5\%$ da média obtida, parâmetro extraído da mesma Fig. 5.2.43, uma tensão acima (ou sob) dessa porcentagem é ruído elétrico.

A Fig. 5.2.45 mostra o esquema utilizado para sintonizar o filtro na caracterização do sensor de distância, as medições foram feitas cada 10 mm, junto com a sintonização do filtro PB utilizando a programação gerada na Fig. 5.2.24.

Figura 5.2.45. Calibração e sintonização do filtro passa baixa digital para o sensor GP2Y0A21YK da SHARP programado no microcontrolador, as medições foram feitas cada 10 mm.



Fonte: Próprio Autor.

O resultado das medições após a sintonização do filtro dentro do microcontrolador é apresentado na Tab. (5.2.10). O microcontrolador utilizado foi o PIC18F14550.

Tabela 5.2.10. Valores obtidos no sensor GP2Y0A21YK após filtragem.

Distância Medida (cm)	Valor no Microcontrolador Após o filtro PB (Inteiro 16 bits)	Distância Medida (cm)	Valor no Microcontrolador Após o filtro PB (Inteiro 16 bits)
7	149	15	298
7,5	153	15,5	310
8	159	16	322
8,5	170	16,5	331
9	180	17	340
9,5	188	17,5	350
10	196	18	360
10,5	205	18,5	369
11	214	19	378
11,5	224	19,5	384
12	235	20	391
12,5	245	20,5	396
13	255	21	402
13,5	268	21,5	409
14	280	22	417
14,5	289	22,5	430

Fonte: Próprio autor.

Os valores de distância calculados internamente no microcontrolador PIC18F4550, alistados na Tab. (5.2.10), foram traçados na Fig. 5.2.56 para observar a linearização do sensor, após a filtragem.

Figura 5.2.46. Gráfica característica do sensor GP2Y0A21YK da SHARP programado no microcontrolador.

Fonte: Próprio Autor.

5.2.5.3 Bloco Driver Bipolar e controle da corrente

Conforme já mencionado anteriormente o driver do motor de passo a ser projetado deve ser bipolar para ganhar em força e torque no uso de motores de passo unipolares. Além desta característica o driver deve poder ser utilizado com motores de passo na faixa de corrente de até 4 A, dado que no momento do projeto não havia um motor de passo definido para a aplicação.

Para facilitar a implantação do sistema foi escolhido o driver em circuito integrado de propósito geral e baixo custo que é o L6203, este é um driver com as características descritas na Tab. (5.2.11).

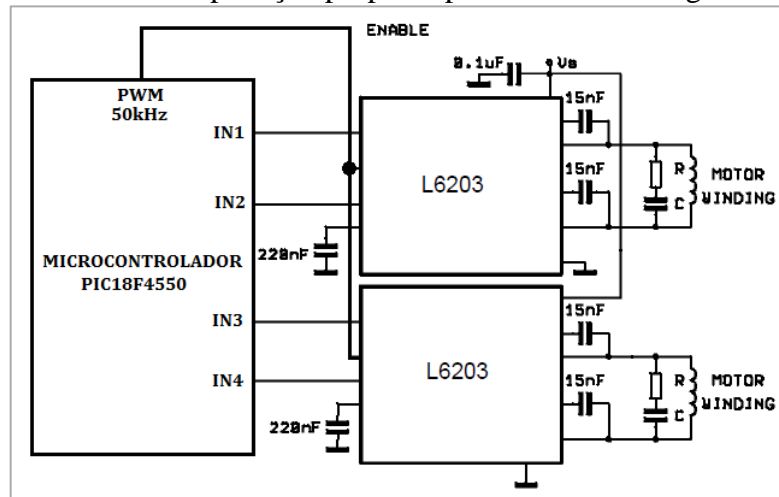
Tabela 5.2.11. Características do driver L6203.

Parâmetro	Valor
Corrente nominal	4A
Compatibilidade	Microcontrolador
Níveis Lógicos	5V
Frequência de Chaveamento	100kHz
Proteção de colisão de saída	Sim

Fonte: DTS (L6203).

O diagrama básico de aplicação do driver sugerido pelo fabricante precisa do uso do circuito integrado L6506, mas isto não é necessário sabendo o princípio de operação do motor de passo e tendo que o microcontrolador possui um modulo interno para geração do PWM, o uso deste circuito integrado auxiliar é dispensado e, um novo circuito de aplicação é proposto e mostrado na Fig. 5.2.47.

Figura 5.2.47. Sistema de aplicação proposto para o circuito integrado driver L6203.



Fonte: Modificado do DTS (L6203).

Os componentes eletrônicos ligados ao circuito (os capacitores de 15 nF, 220 nF e 0,1 μ F), são componentes necessários para o funcionamento dos circuitos internos do driver, um deles é chamado “Charge Pump” que é um circuito desenvolvido internamente para ativar os transistores superiores da ponte H, outro dos subcircuitos internos é o sistema de tensão de referência para a ativação do sistema que gera a lógica interna. Sem esses componentes o driver não funcionaria apropriadamente. Os componentes R e C ao lado do motor são a rede Snubber para proteger o driver contra sobre correntes geradas pelo motor. O cálculo da rede Snubber é mostrado no DTS do driver L6203.

$$R_{SNUBBER} \cong \frac{V_S}{I_p} \quad (5.2.115)$$

$$C_{SNUBBER} = \frac{I_p}{(dv/dt)} \quad (5.2.116)$$

Onde V_S é a tensão da fonte de alimentação, I_p é a corrente de pico do motor e o (dv/dt) é o incremento máximo da tensão de saída que para os motores de passo é tipicamente de 200 V/ μ s segundo o fabricante do circuito integrado L6203.

Para o sistema proposto a corrente de pico será de 4 A, a tensão da fonte é de 24 V, e se utilizará o valor sugerido do dv/dt . O resultado de aplicar estes valores nas equações propostas pelo fabricante é uma resistência **$R_{SNUBBER}$ de 6 Ω** e um capacitor **$C_{SNUBBER}$ de 20 nF**.

Uma grande função que tem o driver L6203 é que permite controlar a corrente dos motores através da técnica de chaveamento PWM, com isto logra-se incrementar o torque instantâneo do motor usando uma fonte de alimentação acima da tensão nominal do motor e controlar a corrente máxima do motor diminuindo o ciclo de trabalho do PWM, isto minimiza o uso de circuitos externos para o controle da corrente dos motores e assim aumentar a potência mecânica do motor.

O PWM é imposto na porta ENABLE do L6203 que neste caso terá uma frequência de chaveamento de 50 kHz que a metade da frequência de chaveamento máxima do L6203. O esquema do PWM já foi explicado nas secções 5.2.3.16 e 5.2.3.17 para o motor de CC e não se falará mais nesta secção.

Figura 5.2.48. Motor de passo MINEBEA vendido no mercado brasileiro.



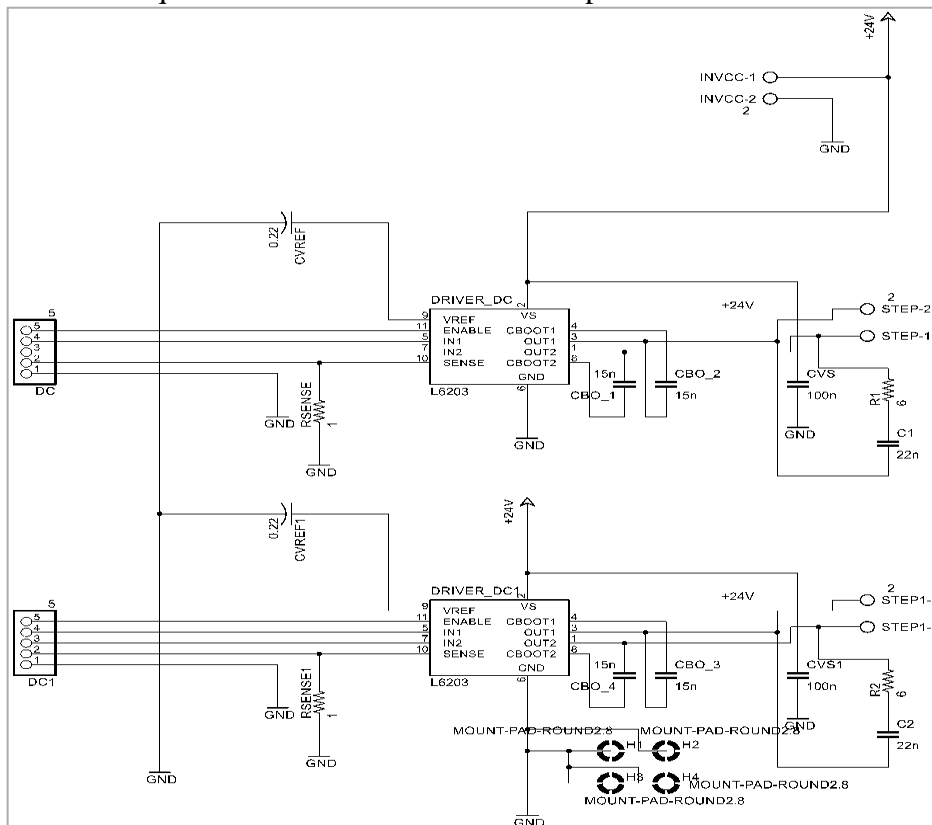
Fonte: Próprio Autor.

A Fig. 5.2.48 mostra o motor de passo comumente utilizado em projetos de controle numérico (CNC) típicos no Brasil, o torque deste motor é de 4 kg/cm em modo unipolar, trocando o modo de operação ao bipolar o torque vai ser em torno dos 7 kg/cm (DTS, MINEBEA 23LM-C004-04). A corrente por fase deste motor de passo é 1,2 A e a tensão de operação deste motor é de 6 V.

Para o motor da Fig. 5.2.48, o ciclo de trabalho do PWM deve ser fixado em 25% para uma tensão da fonte de alimentação de 24 V, com isto o torque instantâneo do motor é 4 vezes o torque nominal. Os fabricantes dos motores de passo recomendam não usar tensões de alimentação superiores ao dobro da tensão nominal do motor, porém o chaveamento aplicado é de 50 kHz e o tempo que a sinal de 24 V é imposta no motor é um 25% do período correspondente à frequência de 50 kHz ou seja 5 μ s, este é um tempo muito curto para danificar o motor pelo que o motor vai operar em segurança, sem ser danificado.

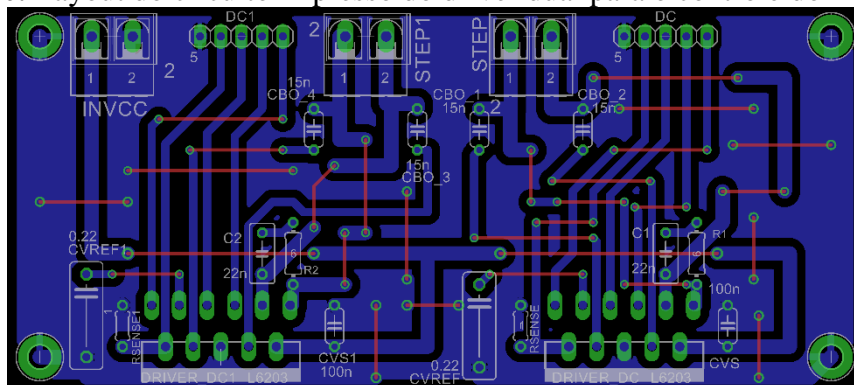
A simulação do circuito integrado não foi realizada porque o dispositivo não conta com um modelo físico-matemático no software de simulação, mas o circuito foi montado e testado. O circuito e layout são apresentados nas Fig. 5.2.49 e Fig. 5.2.50 respectivamente.

Figura 5.2.49. Esquemático do circuito driver dual para o controle do motor de passo.



Fonte: Próprio Autor.

Figura 5.2.50. Layout do circuito impresso do driver dual para o controle do motor de passo.



Fonte: Próprio Autor.

A Fig. 5.2.51 mostra o circuito impresso fabricado após a geração do layout, o circuito impresso montado foi feito de maneira artesanal ao não contar com ferramentas próprias para fabricação de circuitos impressos.

Figura 5.2.51. Circuito impresso do driver dual montado para o controle do motor de passo.



Fonte: Próprio autor.

5.2.5.4 Bloco Sistema Gerador dos Passos

Para o funcionamento dos motores de passo foram programadas no microcontrolador as funções requeridas para a movimentação. O fabricante dos motores MINEBEA sugeriu a sequência típica para o funcionamento dos motores de passo, vista na Fig. 5.2.52.

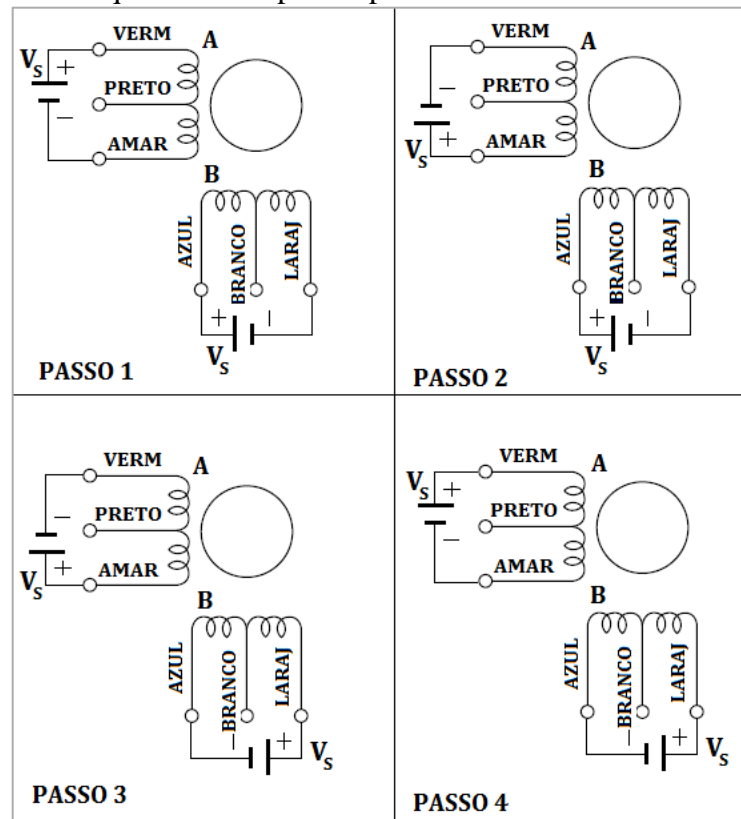
Figura 5.2.52. Sequências dos passos para o controle dos motores MINEBEA.

Bipolar Dual Phase				
Step	A	B	A ¹	B ¹
1	+	+	-	-
2	-	+	+	-
3	-	-	+	+
4	+	-	-	+

Fonte: DTS (MINEBEA 23LM-C004-04).

A sequência da Fig. 5.2.52, é um arranjo básico de pulsos necessários para o funcionamento dos motores de passo MINEBEA, cada passo na Fig. 5.2.52 significa um avanço de 1,8°, no entanto para evitar problemas de ressonância e vibrações do motor nos casos onde o sistema deva dar muitos passos até atingir a posição requerida, vai se propor uma mudança nessa sequência de passos, a justificativa vai se explicar a seguir com ajuda da Fig. 5.2.53.

Figura 5.2.53. Sequências dos passos para o controle dos motores MINEBEA.



Fonte: Modificado do DTS (MINEBEA 23LM-C052-7V).

Observa-se na Fig. 5.2.53 que na mudança de polarização entre o passo 1 e o 2 para a bobina A, acontece num tempo curto uma inversão no fluxo da corrente, dependendo da frequência dos passos esta mudança de polarização pode produzir ressonância no motor chegando a ter vibrações fortes que poderiam causar perda na precisão dos passos e erros no posicionamento do sistema, esse mesmo fenômeno acontece na sequência do passo 2 para o 3, do passo 3 para o 4, etc.

Nas especificações técnicas do circuito integrado L6203, segundo o fabricante o circuito possui internamente uma lógica de proteção para evitar este fenômeno de curto-circuitagem do motor. Muitos autores propõem diferentes sequências de ativação dos motores de passo, no entanto, a sequência proposta neste trabalho, considera o funcionamento eletromagnético dos motores de passo (WILDI, 2007, p. 425-442). A mudança da sequência proposta pelo fabricante na Fig. 5.2.53 por outra sequência de passos intermediários, busca evitar o fenômeno de ressonância. Os passos intermediários é uma pratica altamente recomendável que melhorará a funcionalidade do sistema e aumentará a precisão dos passos.

A nova sequência proposta para evitar a mudança brusca da tensão nas bobinas, é apresentada na Tab. (5.2.12), colocando ao invés das etiquetas A e B, as correspondentes aplicadas na programação no microcontrolador PIC18F4550.

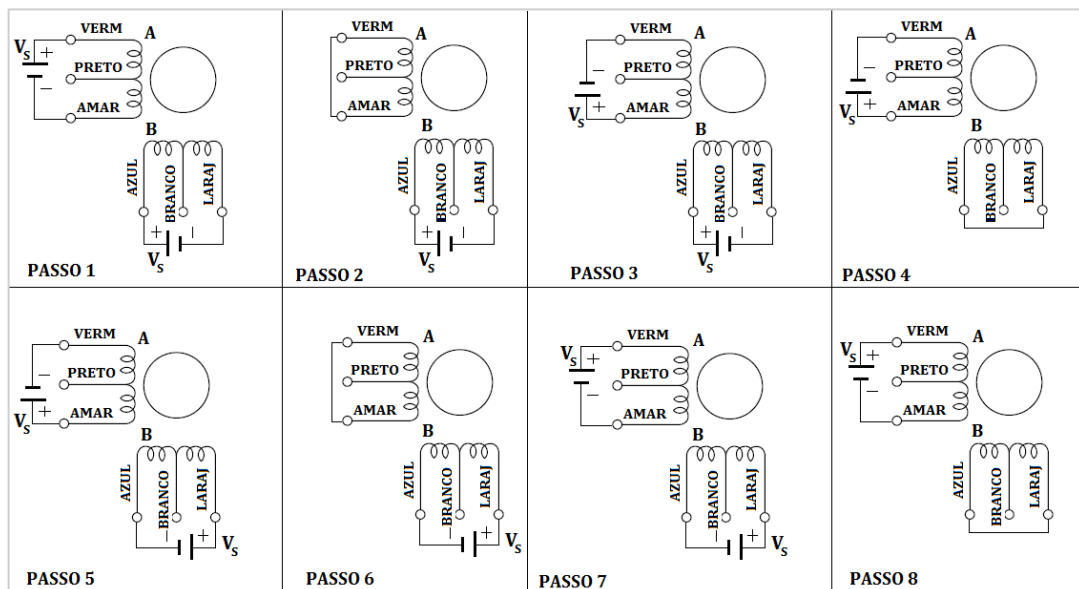
Tabela 5.2.12. Sequência de passos implantada no microcontrolador PIC18F4550 para evitar curto-circuito na saída do L6203.

PASSOS	N1	N2	N3	N4	DEC	BIN
1	5V	0V	5V	0V	10	1010
2	0V	0V	5V	0V	2	0010
3	0V	5V	5V	0V	6	0110
4	0V	5V	0V	0V	4	0100
5	0V	5V	0V	5V	5	0101
6	0V	0V	0V	5V	1	0001
7	5V	0V	0V	5V	9	1001
8	5V	0V	0V	0V	8	1000

Fonte: Próprio autor.

Essa sequência de passos da Tab. (5.2.12) vai gerar a seguinte ativação nas bobinas do motor mostrada na Fig. 5.2.54.

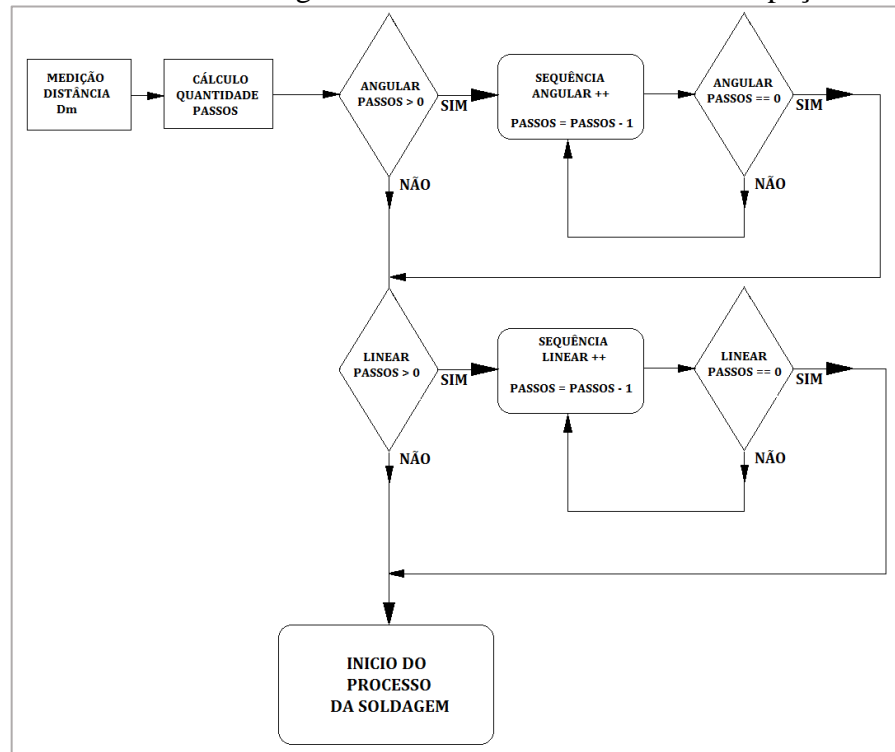
Figura 5.2.54. Sequências dos passos para o controle implantados no microcontrolador PIC18F4550.



Fonte: Modificado do DTS (MINEBEA 23LM-C052-7V).

O processo necessário para o posicionamento da tocha no ponto inicial da soldagem é descrito no fluxograma implantado no microcontrolador para o controle da posição, o qual é apresentado na Fig. 5.2.55.

Figura 5.2.55. Algoritmo computacional programado no PIC18F4550 para o posicionamento da tocha de soldagem baseado na distância da tocha à peça.



Fonte: Próprio Autor.

5.2.6 Sistema de comunicação entre o sistema de posicionamento CNC e o sistema de correção e controle da velocidade constante

Dadas as enormes limitações de orçamento e a dificuldade na compra de componentes e peças eletrônicas nas cidades do interior do sul do Brasil, além das limitações de espaço internamente no robô foi necessário subdividir os sistemas do robô, em dois grandes subsistemas que foram o sistema angular e o sistema de velocidade constante e correção da trajetória.

Os projetistas dos sistemas digitais reconhecem neles três grandes seções que são (TOCCI e WIDMER, 2010, p. 745-765)

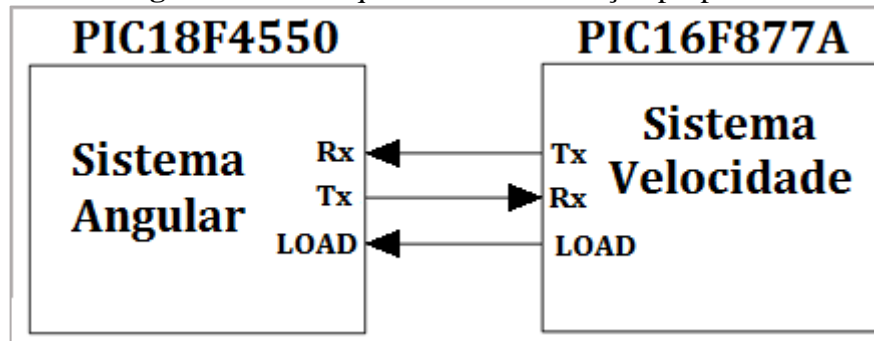
1. Interfaces.
2. Unidade de Processamento Central (UPC)
3. Unidade de sincronização.

As três secções antes mencionadas devem ser projetadas em conjunto, no entanto a unidade de sincronização é a mais importante na hora do projeto e escolha da estrutura final do sistema.

Neste ponto cabe ressaltar que a separação dos subsistemas foi feita pelas múltiplas limitações no início do projeto, mas priorizando a correta sincronização das unidades, uma nova versão do sistema deverá integrar o sistema numa única unidade, sem a necessidade desta interface de comunicação, seguindo as boas práticas de projeto de sistemas digitais. A criação de interfaces de comunicação em alta velocidade deve ser evitada na medida do possível para diminuir a necessidade de protocolos de comunicação e reduzir a complexidade do sistema.

Dadas as limitações de portas para interligação dos sensores, interface com o usuário, e as distâncias de interligação dos sistemas foi criada uma unidade de comunicação serial não síncrona para o envio dos comandos de operação dos subsistemas. O esquema da interligação desta unidade de comunicação apresenta-se na Fig. 5.2.56.

Figura 5.2.56. Esquema de comunicação proposto.

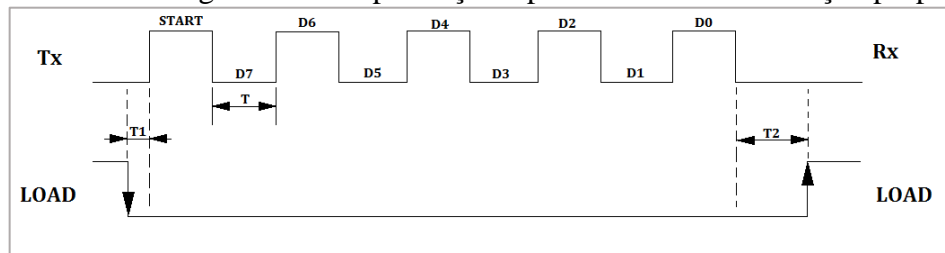


Fonte: Próprio Autor.

A UPC encontra-se dentro do sistema de velocidade-trajetória e é a unidade que define quando e como é estabelecido o envio dos comandos de operação.

O diagrama de temporização e o protocolo de comunicação definido são apresentados na Fig. 5.2.57 (TOCCI e WIDMER, 2010, p. 13).

Figura 5.2.57. Diagrama de temporização e protocolo de comunicação proposto.



Fonte: Próprio Autor.

A Tab. (5.2.13) resume os parâmetros de temporização definidos para o sistema de comunicação.

Tabela 5.2.13. Parâmetros definidos para a temporização na intercomunicação dos subsistemas.

Descrição	Parâmetro	Valor
Tempo de estabilização	T_1	10 μ s
Tempo de parada	T_2	20 μ s
Período	T	20 μ s
Frequência	$1/T$	50 kHz

Fonte: Próprio autor.

Os comandos definidos para a intercomunicação entre os sistemas apresentam-se na Tab. (5.2.14).

Tabela 5.2.14. Comandos definidos para a intercomunicação dos subsistemas.

Comando	Valor Hexadecimal
COMEÇAR	0x09
PARAR	0x05
SUBIR	0x10
DESCER	0x20
RESETAR	0xFF
OFF	0x00

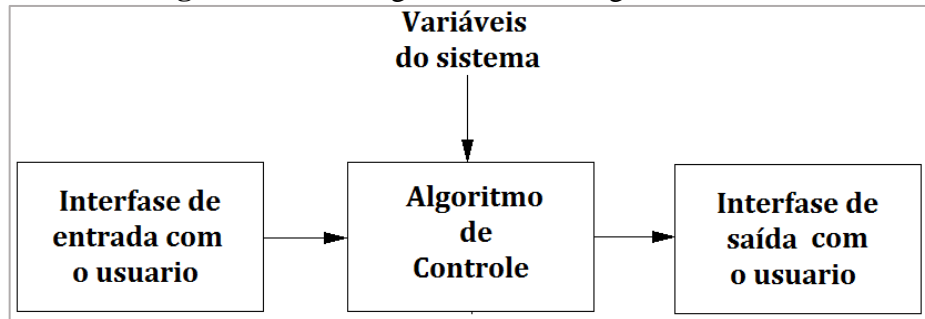
Fonte: Próprio autor.

5.2.7 Bloco Unidade de Processamento Central (UPC)

O principal objetivo da UPC é servir de interface entre o usuário e todos os subsistemas a serem controlados. A conversão dos dados e informações a valores compreensíveis pelos operadores do equipamento é vital para o desenvolvimento da UPC, dado que internamente as variáveis são números binários ou hexadecimais (TOCCI e WIDMER, 2010, p. 4) os quais devem ser arranjados e convertidos em valores do mundo real

para a aplicação de soldagem. O diagrama de blocos desta unidade é apresentado na Fig. 5.2.58.

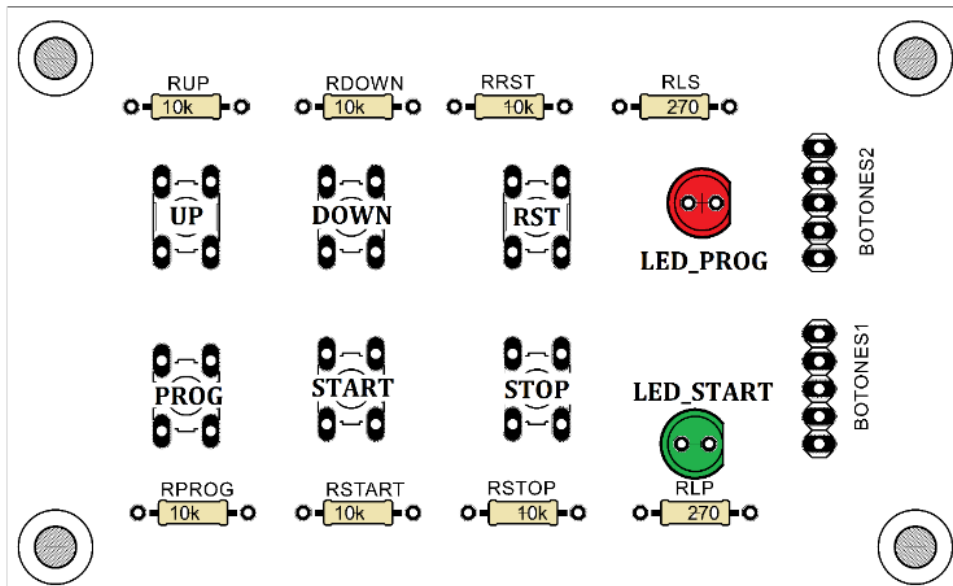
Figura 5.2.58. Diagrama de blocos geral da UPC.



Fonte: Próprio autor.

A interface de entrada com o usuário possui os sinais de entrada mínimos requeridos para definir a velocidade de soldagem que o sistema deve manter constante ao longo do processo. Esses sinais de entrada são apresentados na Fig. 5.2.59 e são explicados na Tab. (5.2.14).

Figura 5.2.59. Interface de usuário definida para o sistema com os sinais necessárias para o correto funcionamento da UPC.



Fonte: Próprio autor.

Tabela 5.2.15. Descrição dos sinais de entrada do sistema para o correto funcionamento da UPC.

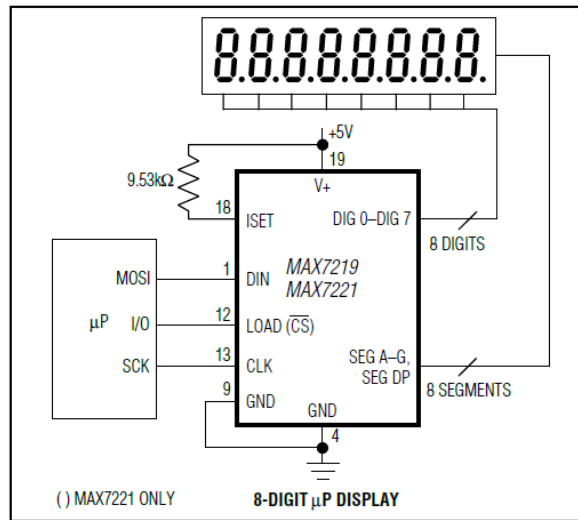
Sinal	Descrição
RST	Sinal de entrada para reiniciar o sistema completo, caso aconteça algum erro geral
PROG	Sinal de entrada para iniciar o processo de programação da velocidade de soldagem
UP	Sinal de entrada para aumentar o valor da velocidade de soldagem a se programar
DOWN	Sinal de entrada para diminuir o valor da velocidade de soldagem a se programar
START	Sinal de entrada para iniciar o processo de soldagem uma vez a velocidade seja programada e a tocha esteja na posição inicial para a soldagem.
STOP	Sinal de entrada fornecida para deter o processo de soldagem.
LED_START	Sinal indicativa da posição do robô e o início do processo
LED_PROG	Sinal indicativa da posição do robô e do início da programação

Fonte: Próprio autor.

A interface de saída com o usuário foi definida como um arranjo de 4 displays de 7 segmentos como unidade mínima para mostrar a velocidade sendo programada e, medida no processo de soldagem com ajuda do encoder. Para facilitar o processo de visualização foi empregado o circuito integrado MAX7219, que é um Driver de displays de 7 segmentos utilizando apenas 3 portas de controle, o que liberaria o microcontrolador PIC16F877A para realizar outras funções além do refrescamento dos dados.

Um componente eletrônico display de 7 segmentos é um dispositivo que apresenta 8 pinos de entrada, se neste caso particular fossem utilizados 4 dispositivos iguais para a visualização da velocidade, serão necessárias 36 portas disponíveis no microcontrolador para a sua interligação. O circuito integrado MAX7219 é um dispositivo auxiliar para interligar até 8 displays de 7 segmentos com um microcontrolador com apenas 3 portas requeridas do lado do PIC16F877A. Este esquema é apresentado na Fig. 5.2.60.

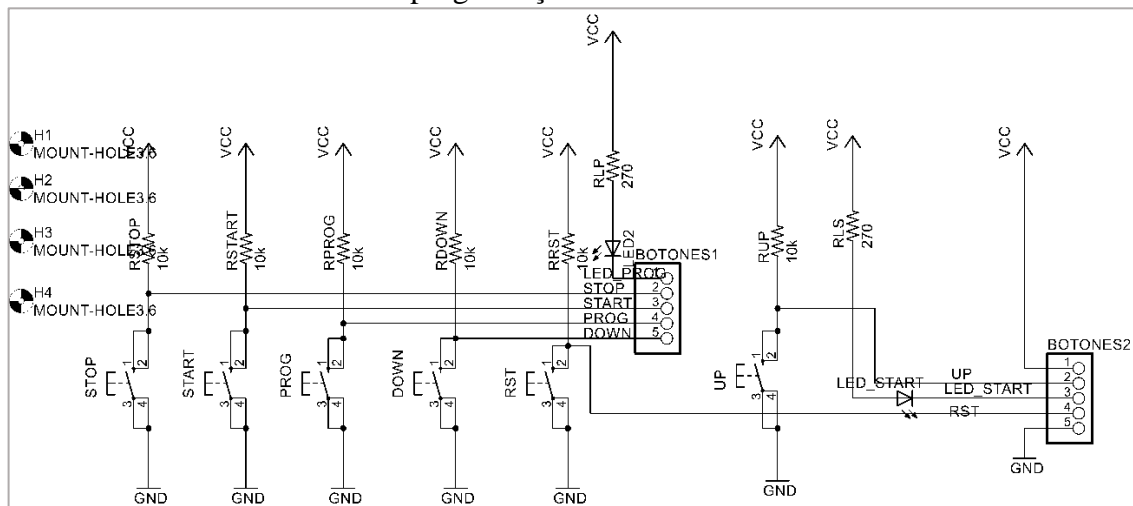
Figura 5.2.60. Diagrama de aplicação típica do MAX7219 fornecido pelo fabricante.



Fonte: DTS (MAX7219).

O circuito implantado usando o software Cadsoft Eagle V5.0, para a interface de entrada do usuário é apresentado na Fig. 5.2.61.

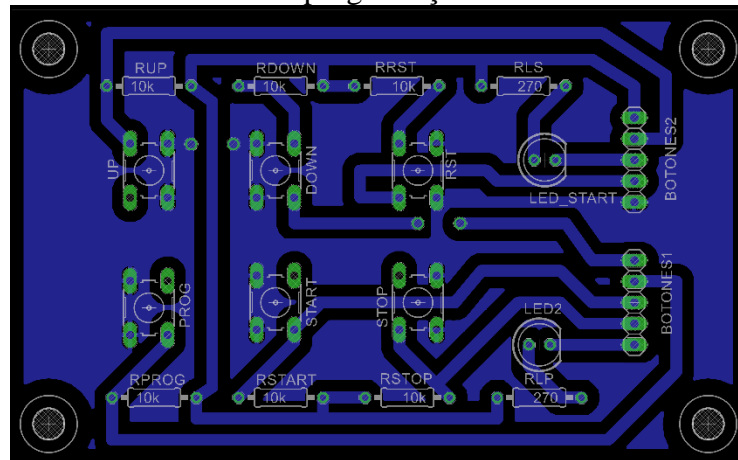
Figura 5.2.61. Circuito da interface de entrada com o usuário para o ingresso da velocidade e programação do sistema.



Fonte: Próprio autor.

O layout do circuito da Fig. 5.2.61 implantado com o software Cadsoft Eagle V5.0, é apresentado na Fig. 5.2.62.

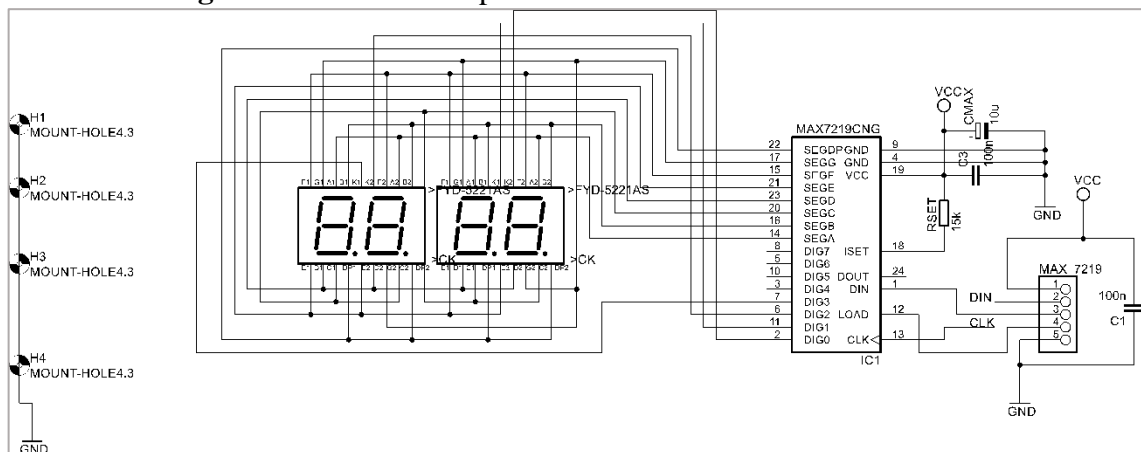
Figura 5.2.62. Layout do Circuito da interface de entrada com o usuário para o ingresso da velocidade e programação do sistema.



Fonte: Próprio usuário.

Como é observado na Fig. 5.2.62, a quantidade de linhas de interligação entre os displays de 7 segmentos e o MAX7219 é de apenas 12. Uma característica interessante do circuito integrado em questão é a habilidade de auto refresco dos dados a ser visualizado, então o dispositivo armazena temporariamente o dado na memória interna que possui até que seja enviado um novo dado a ser visualizado. Isto libera o funcionamento de visualização por parte do microcontrolador, tendo que a UPC somente mandará novos dados quando necessário.

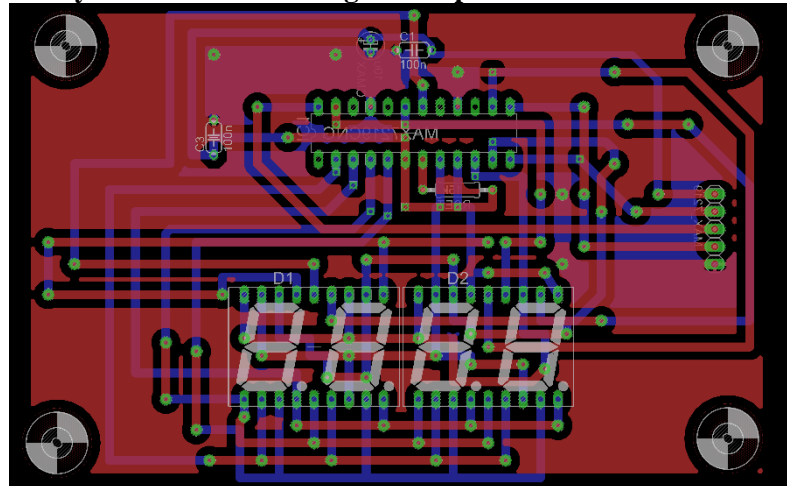
Figura 5.2.63. Circuito para a interface de saída com o usuário.



Fonte: Próprio usuário.

O circuito impresso de visualização de saída foi implantado na tecnologia de dupla face para reduzir o tamanho do circuito impresso, porque como foi dito antes o espaço dentro do robô é limitado. O layout do circuito impresso apresenta-se na Fig. 5.2.64.

Figura 5.2.64. Layout do circuito da Fig. 5.2.62 para a interface de saída com o usuário



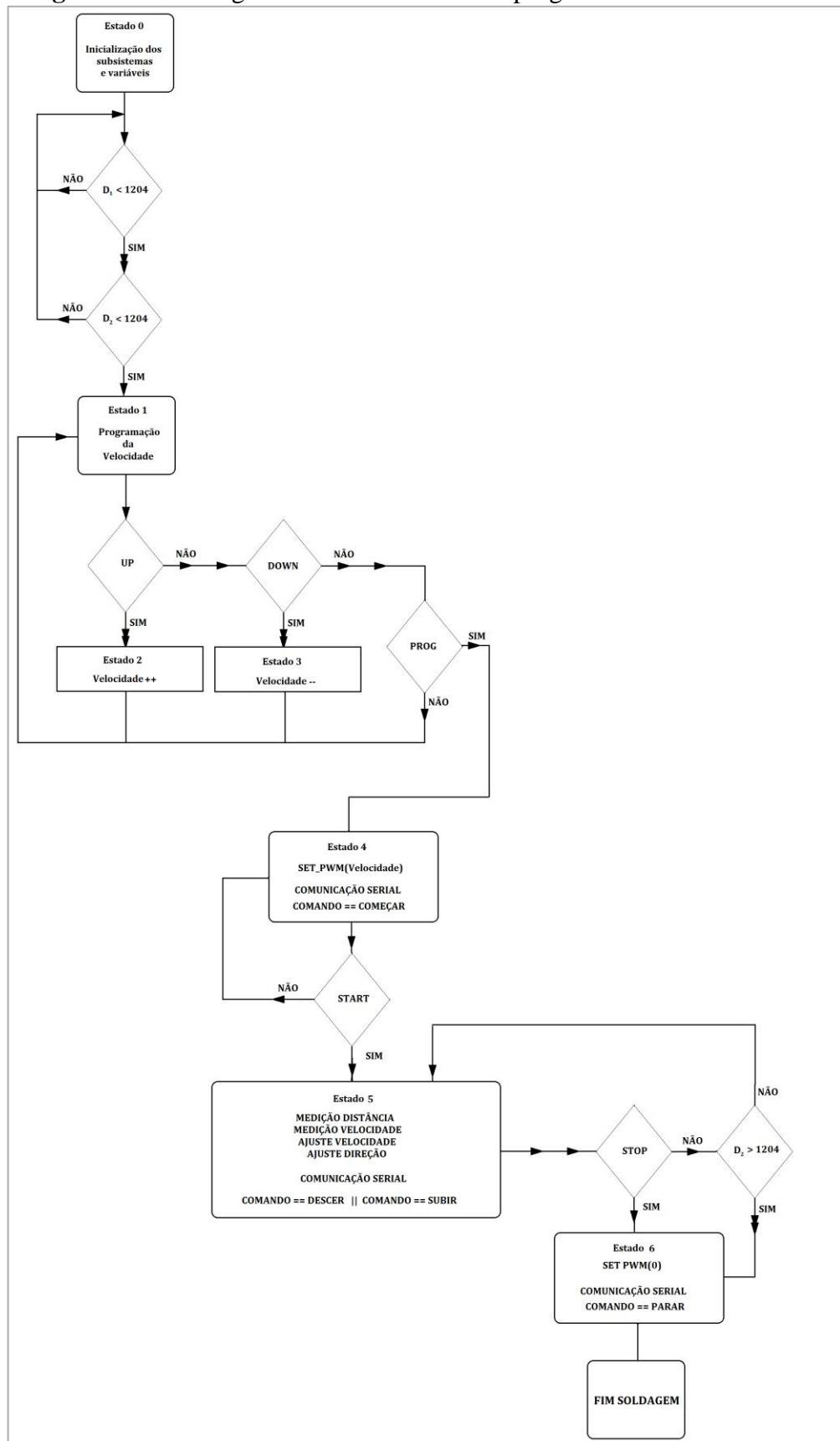
Fonte: Próprio usuário

O diagrama de estados programado para a execução do controle do robô é apresentado na Fig. 5.2.65. O diagrama de estados contém o compilado de funções e processos dos subsistemas projetados e executados no projeto, e que foram demonstrados no Capítulo 5.

As variáveis e algoritmos criados em cada subsistema são chamados na máquina de estados embarcada (MACHADO, 2011) no microcontrolador PIC16F877A. A intercomunicação dos subsistemas programados no PIC16F877A e no microcontrolador PIC18F4550, foi feita através do protocolo de comunicação definido anteriormente.

Todos os algoritmos computacionais programados foram testados com sucesso. O funcionamento do sistema foi aceitável devido a que executou todas as funções como foram programadas, no entanto acredita-se que o sistema ainda deve melhorar tendo que os algoritmos e processos gerados podem ser otimizados.

Figura 5.2.65. Diagrama de estados da UPC programado no PIC16F877A



Fonte: Próprio autor.

Capítulo 6. Produtos Mecânicos e Resultados Experimentais dos Sistemas

6.1 Produtos mecânicos da direção

A peça da Fig. 6.1.1, faz parte do mecanismo da direção, a qual suporta o peso balanceado do mecanismo. Essa peça fornece ao sistema a possibilidade de girar a roda para corrigir a trajetória da soldagem.

Figura 6.1.1. Peça da direção



Fonte: Próprio autor.

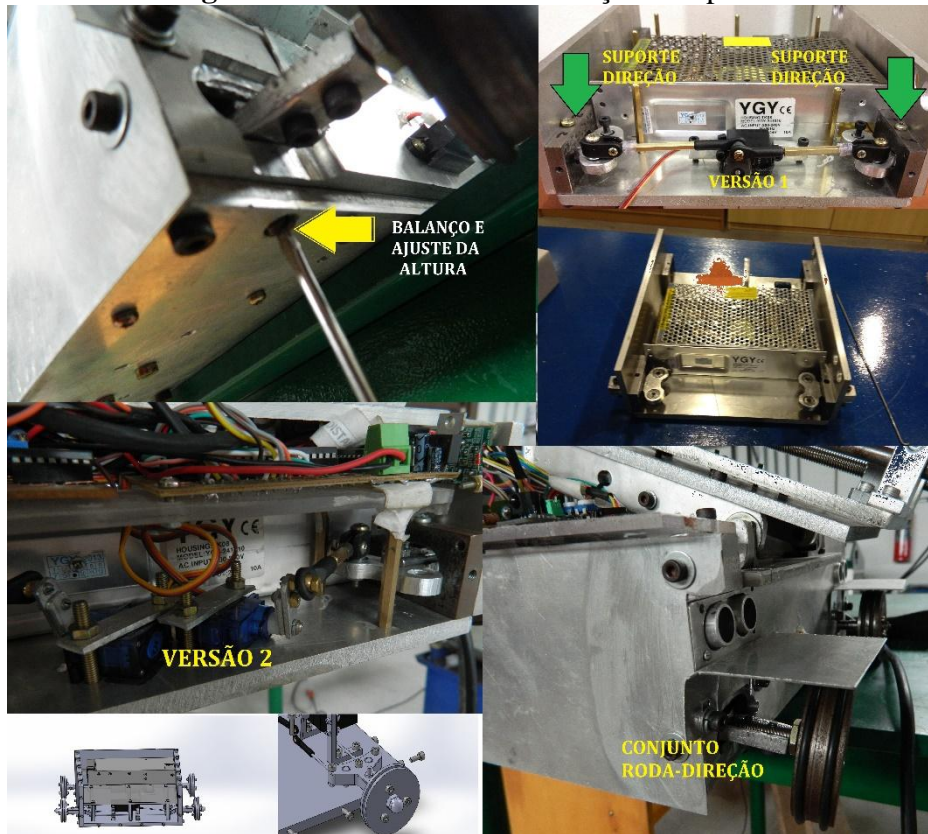
A peça foi feita de alumínio fazendo uso da retifica DREMEL TRIO®, devido às limitações de equipamento de precisão (fresadora), procedeu-se construir manualmente para apressar o processo de testes do sistema.

O resultado obtido foi aceitável, o suficiente para fazer funcionar o mecanismo e testar os algoritmos de controle gerados e programados na UPC.

No entanto, tiveram-se várias complicações de acoplamento do rolamento central (baixa interferência na acoplamento), dado que em certas situações, movia-se para fora do sulco em grandes percursos do carrinho nos testes finais do sistema completo.

A Fig. 6.1.2 apresenta o mecanismo completo colocado dentro de suportes de aço que tem duas funções suportar a estrutura completa e o mecanismo da direção.

Figura 6.1.2. Mecanismo da direção completo



Fonte: Próprio Autor.

Na Fig. 6.1.2, enxerga-se na parte superior esquerda, o mecanismo criado para ajustar e balançar o peso do robô. Esse subsistema mecânico para balançar os esforços foi fundamental para o correto funcionamento do sistema, devido a que os cálculos feitos do torque necessário dos servomotores foram justos, pelo que o mecanismo não funcionava se o sistema não tinha corretamente distribuídas as cargas.

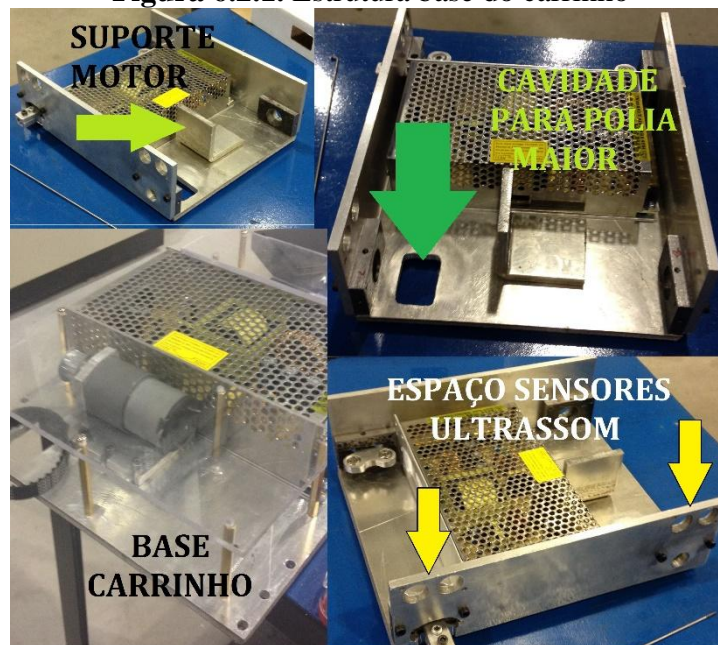
Apresenta-se na figura a primeira versão criada com um único servomotor com o torque requerido, no entanto, devido às imperfeições da fabricação mecânica do robô, não era possível o alinhamento da trajetória reta do robô, pelo que foi necessário colocar dois servomotores com o torque requerido para individualizar o alinhamento de cada roda e atingir a trajetória reta. Na Fig. 6.1.2 se antepõem o desenho feito no Solidworks® e o conjunto roda-direção.

6.2 Produto mecânico carrinho

A estrutura de base do carrinho é mostrada na Fig. 6.2.1, onde foram destacados os pontos a seguir:

- O suporte do motor de CC que movimenta o robô.
- A cavidade para a polia maior que corrige o problema identificado no projeto visto na Fig. 4.2.4.
- A peça base que suporta todo o robô.
- Os espaços criados no projeto para colocação dos sensores por ultrassom US-100.

Figura 6.2.1. Estrutura base do carrinho

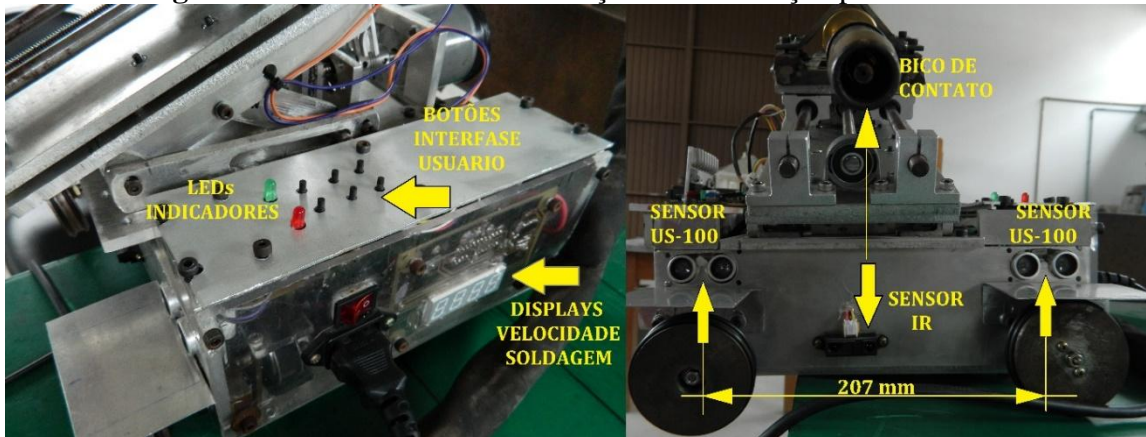


Fonte: Próprio autor.

Dentro da estrutura base considerou-se a colocação estratégica dos sensores, e a colocação da interface de usuário e os botões para programação da velocidade de soldagem. A Fig. 6.2.2 mostram as distâncias medidas na colocação dos sensores de ultrassom e o posicionamento do sensor IR no eixo da tocha de soldagem, isto para medir a distância certa em linha reta até a peça e automatizar o posicionamento da tocha baseado nesta distância.

No carrinho os botões foram colocados na parte traseira acima dos displays de 7 segmentos para facilitar a visualização do usuário ao programar a velocidade e para visualizar os avisos gerados pela UPC no funcionamento do sistema.

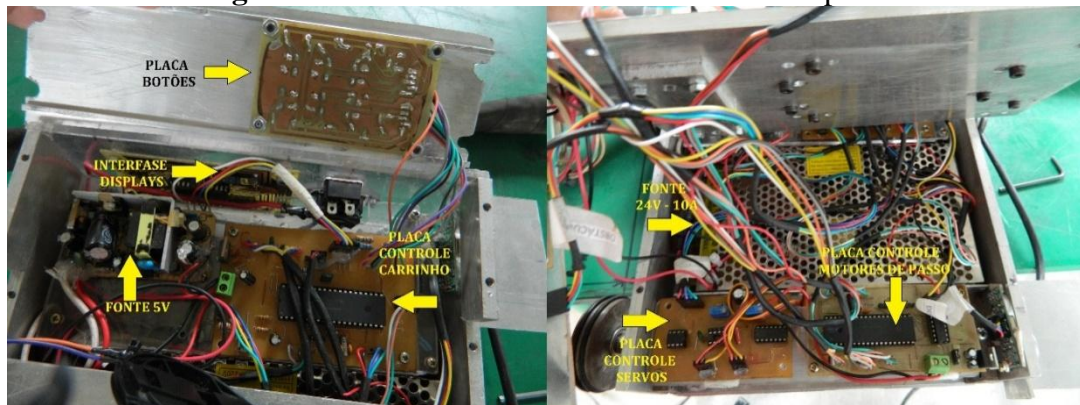
Figura 6.2.2. Mecanismos de inserção de informação para controle



Fonte: Próprio Autor.

A Fig. 6.2.3, apresenta a parte interna do carrinho onde foram colocadas as placas de controle e as fontes de potência do sistema.

Figura 6.2.3. Placas de controle do sistema completo



Fonte: Próprio autor.

Por segurança a carcaça metálica foi ligada a terra, isto para evitar possíveis danos por causa de algum choque elétrico.

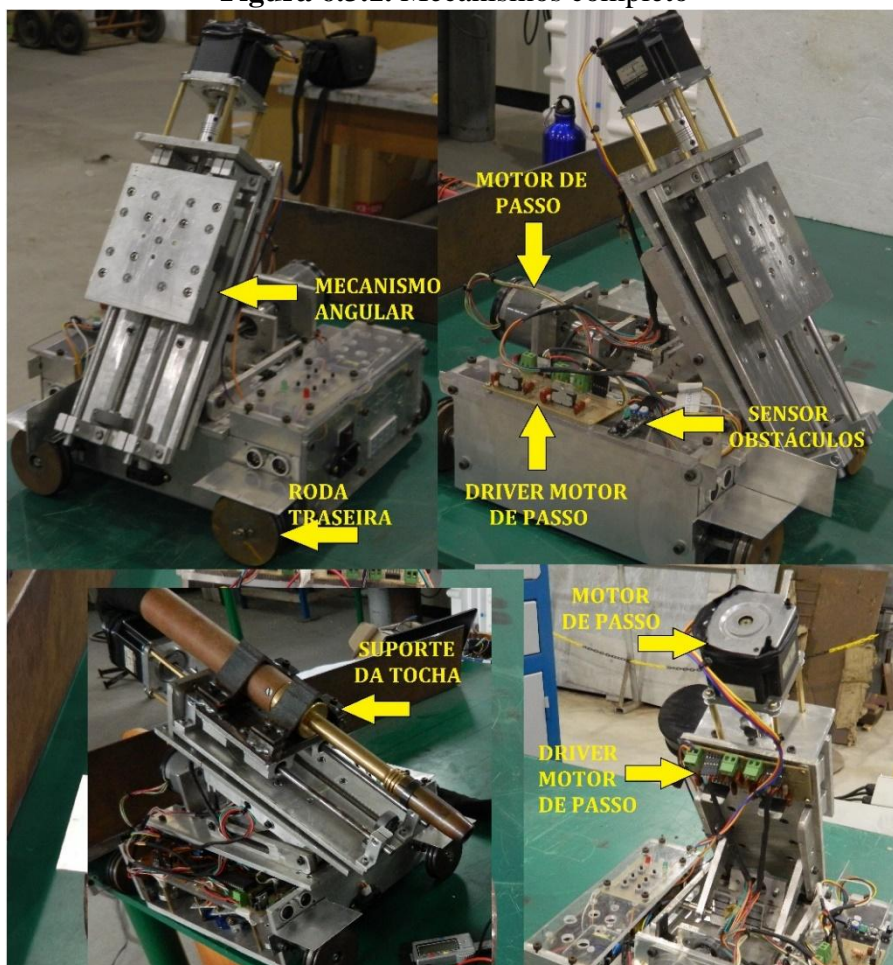
A seguir se alistem os componentes vistos na Fig. 6.2.3.

- Interface dos displays e a placa de controle do carrinho.
- As fontes de 5 V-3 A, para os sistemas digitais, e a fonte de 24 V-10 A para os motores de CC e de passo.
- Placas de controle dos servos-motores, do sistema de posicionamento e de controle do carrinho e da velocidade de soldagem.
- Placa dos botões para programação da velocidade de soldagem.

6.3 Produto mecânico completo

O mecanismo completo é mostrado na Fig. 6.3.1, onde foram etiquetados os componentes principais do sistema.

Figura 6.3.1. Mecanismos completo

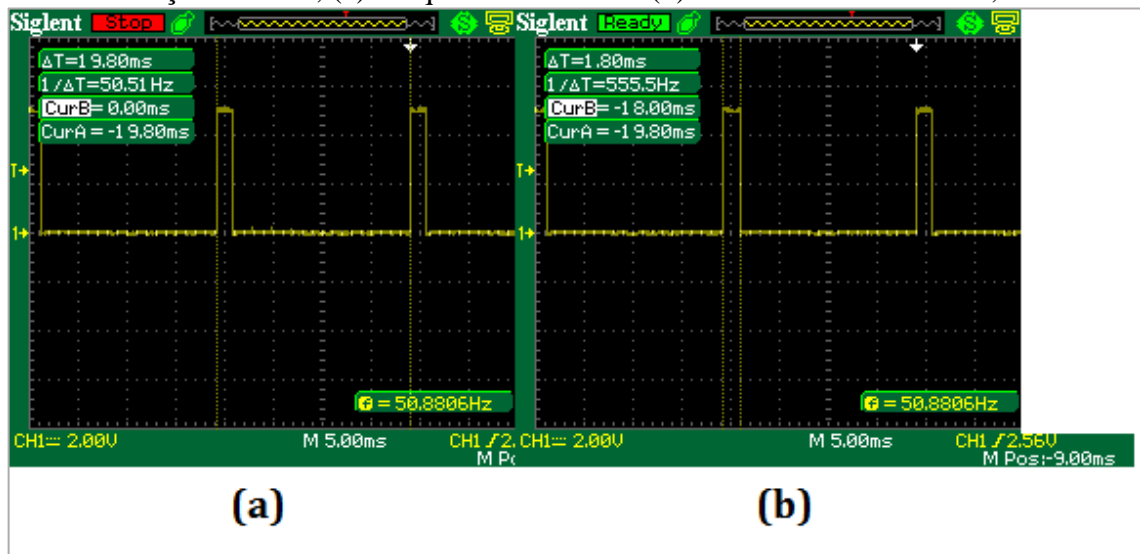


Fonte: Próprio autor.

6.4 Funcionamento dos Servos-motores no sistema da direção.

A Fig. 6.4.1 apresenta o sinal medida com o osciloscópio recebida pelos servos no processo de controle.

Figura 6.4.1. Sinal medida na entrada dos servos gerada pelo PIC10F322 para o controle da direção do robô, (a) Frequência do PWM (b) Ciclo de Trabalho de 9,1%



Fonte: Próprio Autor.

No caso acontecido na Fig. 6.4.1, esse sinal corresponde à fixação da direção no ponto central, quando o carrinho está alinhado com a peça vertical.

6.5 Medições da velocidade do motor aplicando o PWM criado.

O resultado de aplicar a sinal PWM ao motor de corrente contínua, mostra-se na Fig. 6.5.1, para um ciclo de trabalho do 17,6% (proporcional a uma tensão média de 4,22 V). Na sequência de imagens o que se pode enxergar é a saída do sensor de velocidade ao aplicar a tensão do PWM ao motor. A sequência mostra os pulsos gerados no sensor de esquerda para direita e de acima para baixo, a distância percorrida pelo carrinho em cada pulso positivo é aproximadamente 0,9425 cm, tendo que por cada volta da roda do carrinho com raio 3 cm, o sensor gera 20 pulsos, supondo separações idênticas, a velocidade medida será então:

$$V_{el} = \frac{6 \cdot \pi \cdot r \text{ [cm]}}{\text{Tempo} \times \text{pulso [s]}} = \frac{56,55}{\text{Tempo} \times \text{pulso [s]}} \left[\frac{\text{cm}}{\text{min}} \right] \quad (6.5.1)$$

A Tab. (6.5.1) apresenta o resumo dos pulsos medidos do encoder para o cálculo da velocidade dentro do microcontrolador, estes pulsos medidos são mostrados na Fig. 6.5.1.

Tabela 6.5.1. Resumo dos pulsos medidos no sensor de velocidade vistos na Fig. 6.5.1

Pulso	Tempo (s)	Velocidade (cm/min)
1	0,920	51,22
2	0,940	50,13
3	0,880	52,54
4	0,960	49,08
5	0,920	51,21
Média	0,924	50,84

Fonte: Próprio Autor.

Para um ciclo de trabalho de 27%, ou seja, uma tensão de 6,492 V, o resultado da velocidade medida no sensor (encoder) se mostra na Fig. 6.5.2 e o resumo apresentam-se na Tab. (6.5.2).

Figura 6.5.1. Pulsos medidos do sensor de velocidade para um ciclo de trabalho do 17,6%



Fonte: Próprio Autor.

Tabela 6.5.2. Resumo dos pulsos medidos no sensor de velocidade vistos na Fig. 6.5.2.

Pulso	Tempo (s)	Velocidade (cm/min)
1	0,500	94,24
2	0,560	84,14
3	0,540	87,26
4	0,520	90,61
5	0,520	90,61
6	0,540	87,26
7	0,520	90,61
8	0,540	87,26
9	0,500	94,24
Média	0,526	89,58

Fonte: Próprio Autor.

Como foi visto na Tab. (6.5.2) a média da velocidade medida após 9 pulsos do encoder foi de 89,58 cm/min, o valor programado para o percurso foi de 90 cm/min. O erro encontrado na medição foi de um 0,46%.

O Tempo máximo de estabilização apresenta os seguintes valores limites devido à função de ajuste do PWM criada.

Tabela 6.5.3. Valores limites de assentamento do sistema com o tempo de ajuste do ciclo de trabalho.

Velocidade [$\frac{cm}{min}$]	t_{delay} [s]	Distância [cm]
16	3,7	0,945
110	0,5	0,945

Fonte: Próprio Autor.

Sem importar o valor da velocidade de soldagem o sistema precisa percorrer 0,945 cm para ajustar a velocidade isto é o tempo entre pulsos do sensor.

Será necessário encontrar agora o valor desta constante proporcional que ajuste o erro de estado estacionário e reajuste do ciclo de trabalho do PWM.

$$C(s) = erro \quad (6.5.2)$$

Como foi dito antes o compensador $C_{PI}(s)$ encontrado pode ser utilizado para compensar perturbações após a correção da velocidade mudando o ciclo de trabalho do PWM com ajuda do sensor (encoder).

Figura 6.5.2. Pulsos medidos do sensor de velocidade para um ciclo de trabalho do 27%.



Fonte: Próprio Autor.

Para a correção do erro na velocidade em cada pulso do encoder a equação de compensação é proporcional à quantidade ou parcela da velocidade menor medida, desta forma em cada medição da velocidade deve-se calcular o erro como é mostrado a seguir.

$$erro = \frac{Velocidade_{alvo} - Velocidade_{atual}}{Velocidade_{alvo}} \quad (6.5.3)$$

A equação de diferenças do sistema com o cálculo do erro para o novo valor no PWM se mostra na Eq. (6.5.4), essa é a equação experimental obtida que melhor se adaptou no ajuste do PWM para todas as velocidades, a frequência de amostragem depende do atraso do sensor que a sua vez depende da velocidade (CT é o ciclo de trabalho).

$$CT(k) = CT(k - 1) + 0,5 \cdot erro \cdot CT(k - 1) \quad (6.5.4)$$

Como exemplo para uma velocidade medida de 16,5 cm/min e uma velocidade alvo de 18 cm/min

$$erro = \frac{Velocidade_{alvo} - Velocidade_{atual}}{Velocidade_{alvo}} = \frac{18 - 16,5}{18} = 0,083$$

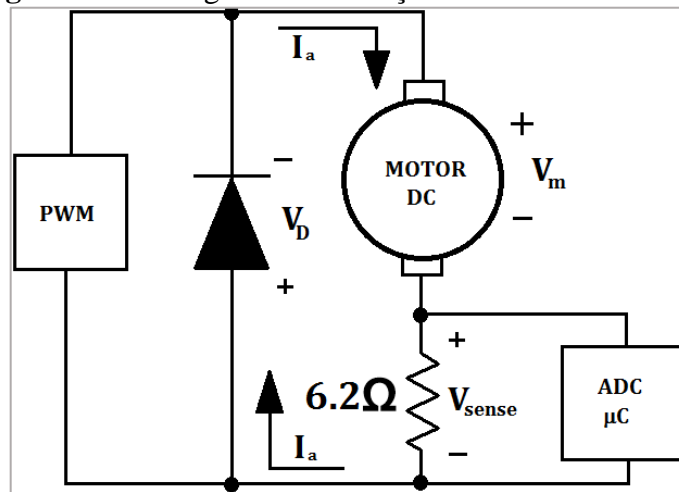
$$CT(k) = CT(16,5) + 0,5 \cdot erro \cdot CT(16,5) = 19,3\% + 0,5 \cdot (0,083 \cdot 19,3\%) = 20,1\%$$

$$CT(k) = 20,1\%$$

Cabe lembrar que isto é graças à linearidade do PWM mantendo o torque instantâneo constante, então uma mudança no torque no motor CC, (KUO, 1996, p. 180), permitirá escalar o novo valor do ciclo de trabalho. Este resultado foi constatado no sistema com o atraso nas medições de velocidade.

6.6 Medição e compensação da corrente da armadura com o microcontrolador

A medição da corrente é feita no microcontrolador através da porta analógica do conversor analógico digital, a frequência de amostragem definida foi de aproximadamente 300 μ s. A fixação da frequência de amostragem do microcontrolador foi definida muito menor do que esse valor para programar um filtro de cálculo da tensão RMS da corrente e escalada usando um resistor de 6,2 Ω .

Figura 6.6.1. Diagrama de medição da corrente do motor.

Fonte: Próprio Autor.

No circuito da Fig. 6.6.1, a escalação da corrente foi feita para aumentar o nível de tensão na entrada do ADC, devido ao baixíssimo nível de corrente máxima do motor de CC para o torque máximo, isto é 150 mA. Ao escalar esta corrente por seis tem-se na entrada do ADC uma tensão próxima a 1 V, mas nunca maior.

O microcontrolador PIC16F877A, permiti uma tensão mínima de entrada na porta analógica de $V_{ss} - 0,3$ V, neste caso V_{ss} é zero, pelo tanto a tensão de entrada mínima na porta ADC, não pode ser menor de 0,3 V negativos.

O caso de maior cuidado é quando o PWM durante o chaveamento vai até zero (PWM OFF), neste ponto o motor vai descarregar a corrente armazenada através do diodo de rolamento livre como se mostra na Fig. 6.6.1 e na porta do ADC se terá a menor tensão.

Esse nível de tensão perigoso se dará no instante imediatamente depois que o PWM apagar, e o seu valor pode se encontrar usando a Lei de Tensões de Kirchhoff (LTK) na malha do circuito da Fig. 6.6.1 (ALEXANDER e SADIKU, 2006, p. 35).

$$V_m - V_D + V_{sense} = 0 \quad (6.6.1)$$

$$V_m = V_{sense} - V_D \quad (6.6.2)$$

A queda de tensão pelo diodo (HER5008) em polarização direta no pior dos casos será de 2 V, o maior nível de corrente (I_a) no circuito corresponderá ao maior torque aplicado ao motor, cujo valor será de 150 mA, o tempo de transição deste diodo é de 250 ns e a tensão instantânea no resistor de medição é

$$V_{sense} = 150 \text{ mA} * 6.2 \Omega = 930 \text{ mV}$$

A tensão instantânea no motor será então imediatamente depois do PWM apagar

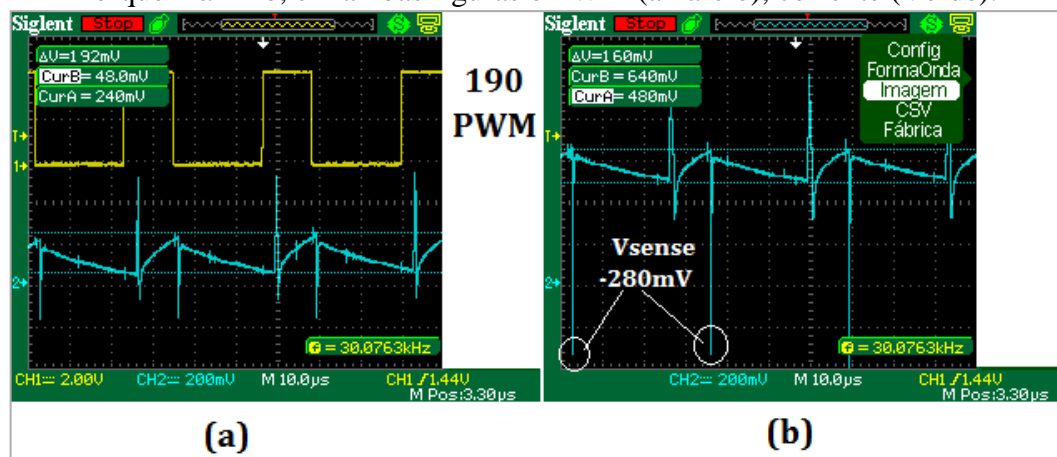
$$V_m = 0,93 V - 2 V = -1,07 V$$

Durante 250 ns a tensão no resistor de medição será

$$V_{sense} = 930 mV - 1,07 = -140 mV$$

No entanto na prática este valor é sempre menor. Este fenômeno é mostrado na Fig. 6.6.2.

Figura 6.6.2. Corrente medida no motor através do resistor R_{sense} . (a) Torque mínimo (b) Torque máximo, em ambas figuras o PWM (amarelo), corrente (Verde).



Fonte: Próprio Autor.

Como pode se enxergar na imagem 6.6.2(b) o valor pico da tensão no resistor na entrada do ADC é próximo aos -300 mV tendo o torque máximo aplicado ao motor (travamento do motor por carga imposta e rotor quase parado). Este fenômeno acontece por uns 250 ns (a frequência do PWM é de 30 kHz).

O intuito desta secção é demonstrar como foi medida a corrente para logo depois proceder com a compensação da corrente de armadura para o controle da velocidade, a frequência de aquisição do ADC definida para esta variável foi de $2 \mu s$, o que permitirá adquirir em torno de 128 amostras da corrente e calcular a tensão média dentro do microcontrolador e gerar uma saída cada $300 \mu s$ (aproximadamente), que foi a frequência de amostragem definida.

Utilizando de novo as medições feitas para a corrente do motor mostradas na Fig. 6.6.2(a), para um ciclo de trabalho do PWM (dentro do microcontrolador) de 190, equivalente ao 17,6%, a velocidade medida com o encoder foi de 50,84 cm/min (resultado da Tab. (6.5.1)), logo depois foi imposto um torque muito maior nas rodas do carrinho e

imediatamente depois foi medida a corrente no sensor (a velocidade já tinha sido fixada em 50 cm/min) a velocidade naturalmente tender-se-á diminuir num valor proporcionalmente menor ao torque imposto. Neste ponto o compensador de corrente deve entrar em operação através da equação de diferenças programada, isto será mostrado a seguir.

Como é sabida, a equação de diferenças obtida para o motor de corrente continua foi:

$$m(k) = 0,09 \cdot e(k) + m(k - 1) \quad (6.6.3)$$

O erro na medição de corrente $e(k)$ segundo a Fig. 6.6.2 é de:

$$e(k) = V_{sense} - V_{anterior} = 640 \text{ mV} - 240 \text{ mV} = 400 \text{ mV} \quad (6.6.4)$$

Usaram-se tensões porque as correntes são todas escaladas pelo resistor de 6.2 Ω , no entanto o delta de corrente interessa em percentagem porque será aplicado ao ciclo de trabalho do PWM.

Dentro do microcontrolador o ADC apresenta uma resolução de 10 bits, que é o mesmo que 1024 valores desde zero até a tensão de alimentação. Isto quer dizer que cada 4,98 mV equivale a um incremento de um bit na medição, então para um valor medido de 4,98 mV ou menor, o valor dentro do microcontrolador será 1, e para um valor medido de 5,1 V o valor será de 1024. No caso dos valores medidos de corrente vistos na Fig. 6.6.2(a) e (b) os valores dentro do microcontrolador serão:

$$e(k) = V_{sense} - V_{anterior} = 128 - 48 = 80$$

Os valores são arredondados porque o microcontrolador não aceita valores em ponto flutuante (ao lado direito do ponto decimal) para variáveis do tipo inteiro. O valor necessário a ser calculado deve ser percentual porque vai se integrar no cálculo do ciclo de trabalho no PWM, então:

$$e(k\%) = \frac{1000 * e(k)}{V_{anterior}} = \frac{1000 * 80}{48} = \frac{80.000}{48} = 1666 \quad (6.6.5)$$

Um fato importante nos cálculos dentro do microcontrolador é que sempre foram utilizadas variáveis do tipo inteiro para garantir uma velocidade de operação alta dentro do dispositivo, por isso todos os cálculos deviam ser escalados, no caso da percentagem de corrente o valor da escala foi 1000, para não perder precisão nas percentagens próximas a 0,5%. O resultado anterior quer dizer que o torque imposto é 166% maior que o torque inicial. A amostra inicial no cálculo é o valor do ciclo de trabalho atual, então:

$$m(k - 1) = DC_{PWM} = 190 \quad (6.6.6)$$

Agora que temos o delta do torque imposto precisamos calcular o delta do ciclo de trabalho que se deve impor para aumentar a velocidade, no entanto para não perder resolução no ciclo de trabalho será necessário escalá-lo antes do cálculo então:

$$m(k - 1) = DC_{PWM} * 10 = 1900 \quad (6.6.7)$$

$$e(k)_{PWM} = DC_{PWM} * 10 * e(k\%) = 1900 * 1666 = 3.165.400$$

O novo ciclo de trabalho com os resultados obtidos é calculado através de operações matemáticas intermediárias devido a que as variáveis internas no microcontrolador são inteiros com signo de 32 bits, isto quer dizer que pode tomar valores:

$$-2.147.483.647 < \text{sign integer 32 bits} < 2.147.483.647$$

Esses processos intermediários no processo de cálculo do novo valor do PWM evitaram o overflow das variáveis, essas operações são mostradas a seguir.

$$Aux = \frac{e(k)_{PWM}}{100} = 31.654 \quad (6.6.8)$$

$$Aux_2 = Aux \times 9 = 284.886 \quad (6.6.9)$$

$$m(k) = \frac{Aux_2}{1.000} + m(k - 1) = \frac{284.886}{1.000} + 1900 \quad (6.6.10)$$

$$m(k) = \frac{284.886}{1.000} + 1900 \quad (6.6.11)$$

$$m(k) = 284 + 1900 = 2184 \quad (6.6.12)$$

Finalmente o novo ciclo de trabalho a se impor na saída do gerador de PWM é

$$m(k) = \frac{m(k)}{10} = 218$$

Isto vai impor na saída uma tensão no motor de

$$Tensão_{PWM_{nova}} = Ciclo_{PWM} * \frac{V_{24}}{1024} = 218 * \frac{24V}{1024} = 5,11 V \quad (6.6.13)$$

$$Tensão_{PWM_{anterior}} = 190 * \frac{24V}{1024} = 4,45 V \quad (6.6.14)$$

$$\Delta T = Tensão_{PWM_{nova}} - Tensão_{PWM_{anterior}} = 5,11 - 4,45 = 0,66 \quad (6.6.15)$$

Ao mesmo tempo vai se refletir numa tensão no resistor 6.2 Ω da seguinte forma

$$V_{sense} = 0,640 - \frac{R_{sense} * \Delta T}{R_{sense} + R_a} = 0,640 - \frac{6,2 * 0,66}{6,2 + 24} = 0,504 \quad (6.6.16)$$

$$V_{anterior} = \frac{R_{sense} * \Delta T}{R_{sense} + R_a} + V_{sense_anterior} = \frac{6,2\Omega * 0,66V}{6,2\Omega + 24\Omega} + 0,240 = 0,375 V$$

Dentro do microcontrolador

$$V_{sense} = \frac{0,504V}{0,00498} = 101$$

$$V_{anterior} = \frac{0,375V}{0,00498} = 75$$

O novo erro será então calculado a seguir.

$$e(k) = V_{sense} - V_{anterior} = 101 - 75 = 26$$

$$e(k\%) = \frac{1000 * e(k)}{corrente_{anterior}} = \frac{1000 * 26}{75} = \frac{26.000}{75} = 346$$

Isto vai se repetir até o erro porcentual se tornar zero, quando isto acontecer a velocidade será de novo a velocidade estabelecida, mas manejando esta nova carga. O processo de compensação apresenta-se nas Tab. (6.6.1) e Tab. (6.6.2).

Tabela 6.6.1. Resumo do algoritmo de cálculo da equação de diferenças dentro do microcontrolador para compensação nas variações da corrente, parte 1.

Amostra	Tempo amostragem (μs)	Corrente (mV)	Valor corrente (μC)	Corrente anterior (mV)	Valor anterior (μC)	e(k%)	m(k - 1)
1	0	640	128	240	48	1666	1900
2	300	504	101	376	75	346	2184
3	600	470	94	409	82	146	2252
4	900	456	91	423	85	70	2281
5	1200	449	90	430	86	46	2295
6	1500	445	89	434	87	22	2304
7	1800	443	89	436	87	22	2308
8	2100	441	88	438	87	11	2312
9	2400	440	88	439	88	0	2314

Fonte: Próprio autor.

Tabela 6.6.2. Continuação da Tab. (6.6.1).

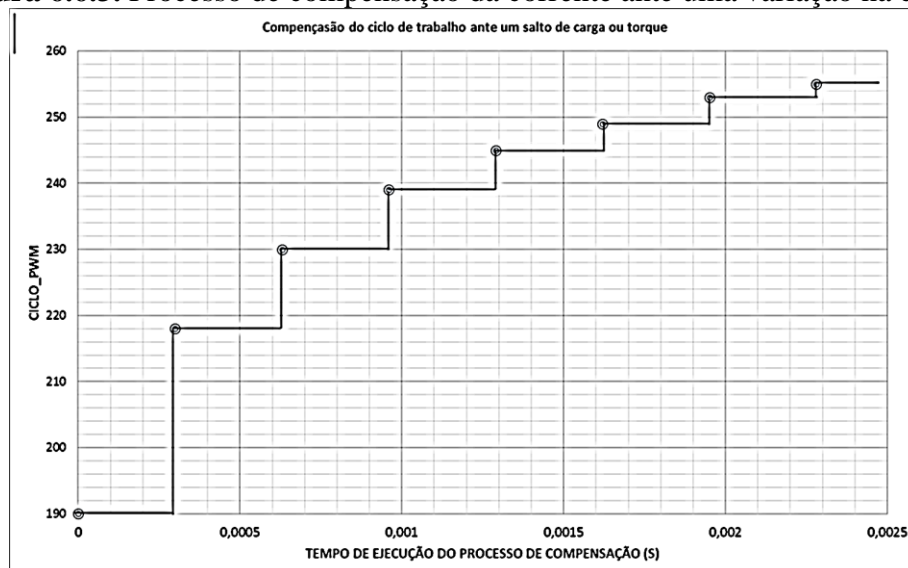
Amostra	Tempo amostragem (μ s)	$e(k)_{PWM}$	Aux	Aux_2	$\frac{Aux_2}{1.000}$	CT	VPWM (V)
1	0	3165400	31654	284886	284	190	4,453
2	300	755664	7556	68004	68	218	5,118
3	600	328792	3287	29583	29	225	5,278
4	900	159670	1596	14364	14	228	5,346
5	1200	105570	1055	9495	9	229	5,378
6	1500	50688	506	4554	4	230	5,400
7	1800	50776	507	4563	4	230	5,409
8	2100	25432	254	2286	2	231	5,418
9	2400	0	0	0	0	231	5,423

Fonte: Próprio autor.

A Tab. (6.6.1) e a Tab. (6.6.2) resumem detalhadamente o processo de compensação dentro do microcontrolador com as escalagens necessárias para não perder precisão pelo truncamento das variáveis no processo de cálculo interno, uma das principais razões pelas quais o sistema pode entrar em estado de instabilidade. Após o processo apresentado nas Tab. (6.6.1) e Tab. (6.6.2) o sistema está pronto para compensar uma nova variação de carga.

Na Fig. 6.6.3 foram expostos os resultados das mudanças do ciclo de trabalho no PWM em cada passo da compensação até que a função do erro seja zero. O tempo final do processo é de aproximadamente 2400 μ s.

Figura 6.6.3. Processo de compensação da corrente ante uma variação na carga.



Fonte: Próprio Autor.

6.7 Comprovação da precisão dos passos do mecanismo angular

O sistema angular foi mandado dar 7600 passos o equivalente a 38 mm, para um tamanho de passo do fuso de 5 mm, o motor de passo dá um giro de 360° cada 200 passos. Para o teste do sistema foram medidos com ajuda da régua os ângulos do mecanismo antes e depois da operação, os valores obtidos se mostram a seguir.

Ângulo inicial = 21 graus

Ângulo Final = 57 graus

Das equações encontradas teoricamente temos que para um ângulo inicial de 57 graus o valor da função cosseno é:

$$\cos(\beta = 57^\circ) = 0,544$$

Os parâmetros medidos no carrinho para as variáveis a, b e c são:

$$a = 68\text{mm}$$

$$b = 57\text{mm}$$

$$c = 118\text{mm}$$

O deslocamento linear total do sistema angular é calculado com a Eq. (5.2.112)

$$\Delta x' = 118 - 57 \cdot \cos(\beta) - \sqrt{(118 - 57 \cdot \cos(\beta))^2 + 13450 \cdot \cos(\beta) - 12550}$$

$$\Delta x' = 38,67 \text{ mm}$$

O resultado do sistema angular ao mandar o motor a dar 7600 passos foi de aproximadamente 38 mm como se mostra na Fig. 6.7.1.

Figura 6.7.1. Deslocamento linear do sistema angular após 7600 passos do motor, o valor medido foi de aproximadamente 38 mm



Fonte: Próprio Autor.

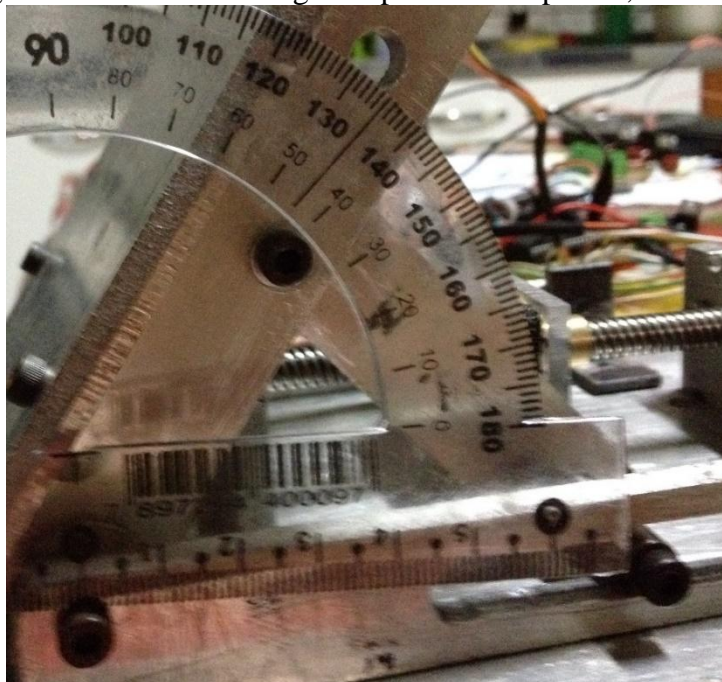
Figura 6.7.2. Ângulo inicial do sistema angular antes os 7600 passos, o valor medido foi de 21°



Fonte: Próprio Autor.

A Fig. 6.7.3 mostra a boa precisão dos passos gerados pelo sistema de posicionamento CNC e pode ser empregada para a colocação da tocha de soldagem baseado na distância do robô à peça vertical a ser soldada.

Figura 6.7.3. Ângulo final do sistema angular após os 7600 passos, o valor medido foi de 57°



Fonte: Próprio Autor.

6.8 Comprovação do ângulo final versus a distância medida à peça

Para uma distância medida de 198 mm os cálculos teóricos com as equações encontradas se apresentam a seguir, no caso da Eq. (5.2.104) e a Eq. (5.2.105).

$$\cos(\beta) = \frac{\sqrt{\left(\frac{17800}{11450 + D_m^2}\right)^2 + 4\left(\frac{D_m^2 - 6972}{11450 + D_m^2}\right) - \left(\frac{17800}{11450 + D_m^2}\right)}}{2} = 0,719$$

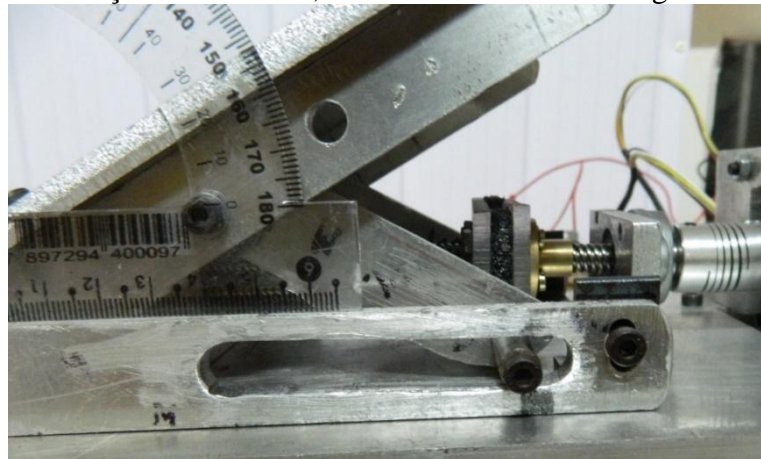
$$\sin(\beta) = \sqrt{1 - (0,704)^2} = 0,694$$

O ângulo inicial da soldagem será então $\beta = 45,25^\circ$ e o cálculo teórico do deslocamento necessário para atingir esse ângulo é calculado com a Eq. (5.2.112) será

$$\Delta x' = 118 - 40,128 - \sqrt{(113 - 40,128)^2 + 9468,8 - 11394} = 14,7 \text{ mm}$$

As imagens a seguir mostram os resultados do sistema angular ao medir uma distância de 200 mm desde o carrinho à peça.

Figura 6.8.1. Ângulo inicial do sistema angular antes de iniciar o deslocamento após a medição da distância, o valor medido foi de 27 graus.



Fonte: Próprio Autor.

A Fig. 6.8.1, mostra o ângulo inicial do sistema que é de 27 graus, o ângulo inicial foi estabelecido nesse valor para calibrar o erro final.

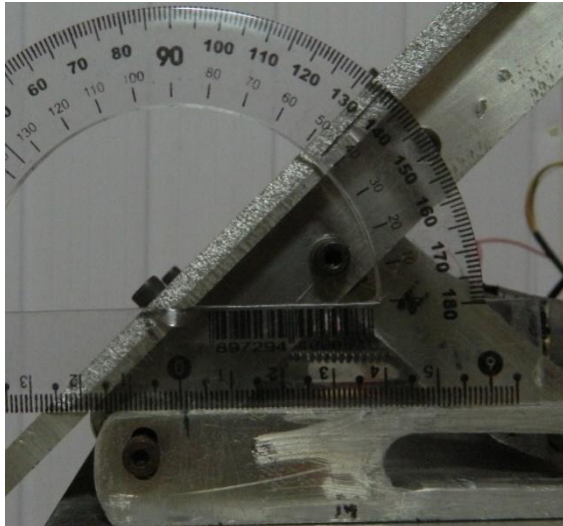
Figura 6.8.2. Distância percorrida pelo sistema para atingir o ângulo final (ou o ângulo inicial da soldagem), o valor medido foi de entre 14 e 15 mm



Fonte: Próprio Autor

Na Fig. 6.8.2, observa-se a distância percorrida pelo braço linear que vai estabelecer o ângulo inicial da tocha de soldagem. Como foi dito antes a distância fixada a ser medida pelo sensor foi de aproximadamente 200 mm, no cálculo teórico o valor dado foi de 14,7 mm, o valor medido visto na Fig. 6.8.2 foi entre 14 e 15 mm. O ângulo atingido com esse deslocamento é visto na Fig. 6.8.3, que se apresenta na Fig. 6.8.3.

Figura 6.8.3. Ângulo final (ou o ângulo inicial da soldagem) atingido após o deslocamento de entre 14 e 15 mm, o valor medido foi de entre 45° e 46°



Fonte: Próprio Autor.

Como visto o sistema apresenta um erro linear de $\pm 0,5$ mm e angular de $\pm 0,5^\circ$, aproximadamente em relação com os dados teóricos calculados.

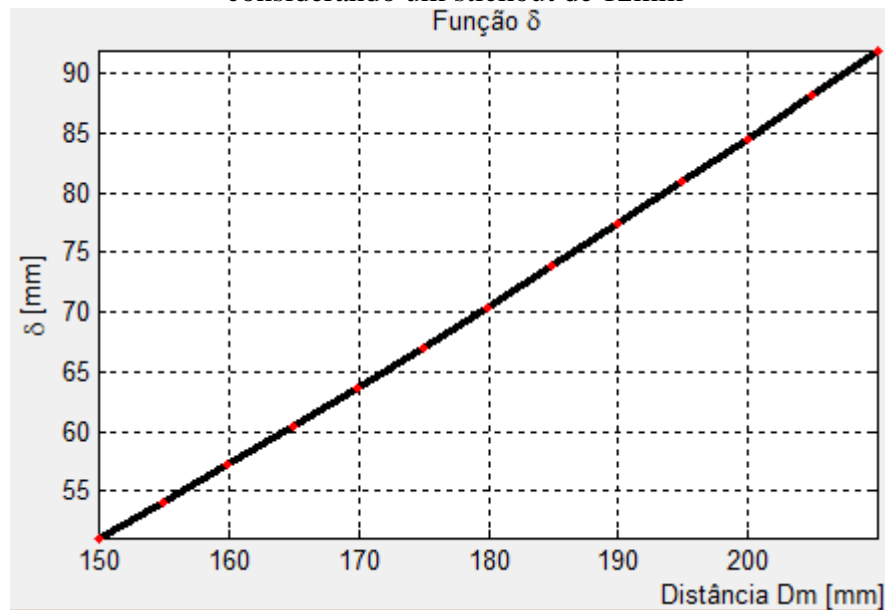
6.9 Comprovação da precisão dos passos do mecanismo linear

Para a comprovação do deslocamento linear este é calculado baseado na Eq. (5.2.114).

$$\delta = \frac{D_m - 119 \cdot \cos(\beta) - 46 \cdot \sin(\beta) - 11 \frac{\cos^2(\beta)}{\sin(\beta)}}{0,93 \cdot \cos(\beta) + 0,36 \cdot \sin(\beta)}$$

A curva traçada para a equação anterior pode ser utilizada para comprovar os passos do sistema linear. A função deslocamento linear da mesa que suporta a tocha é apresentada na Fig. 6.9.1 para diferentes distâncias (sabendo que a função cosseno do ângulo de soldagem depende desta distância).

Figura 6.9.1. Deslocamento linear do mecanismo que suporta a tocha experimental considerando um stickout de 12mm



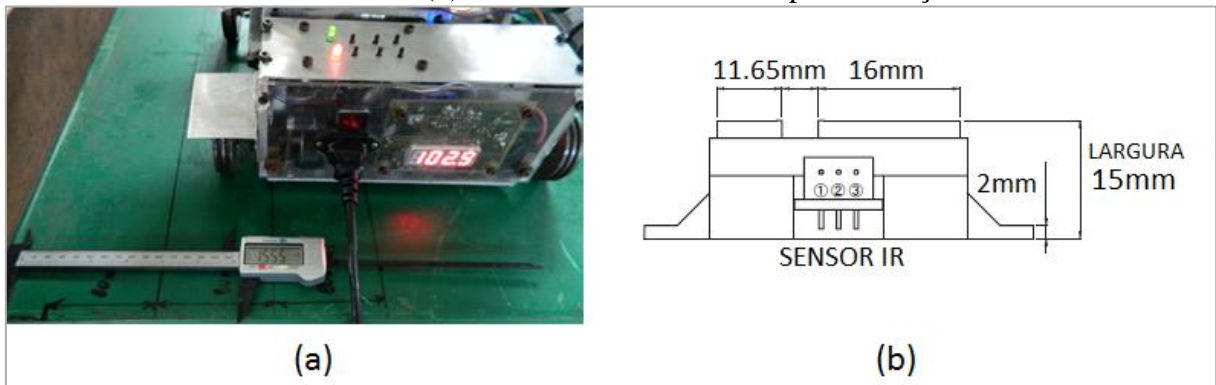
Fonte: Próprio autor

Na Fig. 6.9.2(a), colocou-se o paquímetro para medir a distância do carrinho até a peça, além se tem a medição da distância pelos sensores de ultrassom, cujo valor visto de distância é proporcional a Eq. (5.2.2) (desconsiderar o ponto decimal).

$$Distancia_m = Displays \times 0,167 [mm] = 1029 * 0,167 = 171,8 mm \quad (6.9.1)$$

Pelo posicionamento dos sensores de ultrassom em relação com o sensor IR, internamente no microcontrolador essa distância é corrigida pelo formato físico do sensor IR de distância como se mostra na Fig. 6.9.2 (b) e baseado no centro instantâneo do mecanismo.

Figura 6.9.2. Distância medida do carrinho à peça (a) Medição com o paquímetro e no sensor de ultrassom (b) Dimensões do sensor IR para correção



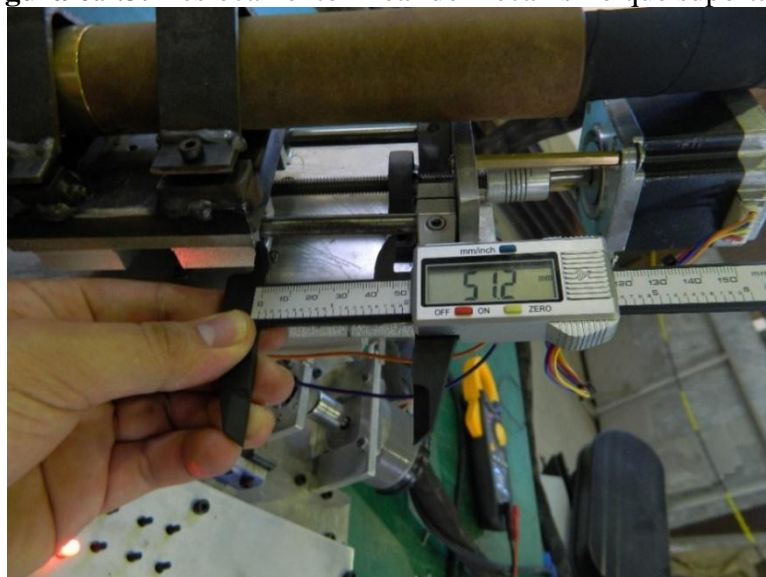
Fonte: Próprio autor.

O centro instantâneo pode ser visto na Fig. 5.2.36 e ao código programado para o μC 18F4550 no apêndice (A). Então a distância para o cálculo internamente utilizando o resultado da Eq. (6.9.1) é:

$$D_m = Distancia_m - 15mm = 156,8 mm \quad (6.9.2)$$

Esses 15 mm na Eq. (6.9.2) são pelo formato físico do sensor IR devido à largura do mesmo como enxergado na Fig. 6.9.2(b). Segundo a Fig. 6.9.1 para um D_m de 156,8 mm, corresponde um deslocamento (δ) de 53 mm, esse resultado apresenta um erro de 2 mm (aproximadamente) como foi medido segundo a Fig. 6.9.3.

Figura 6.9.3. Deslocamento linear do mecanismo que suporta a tocha

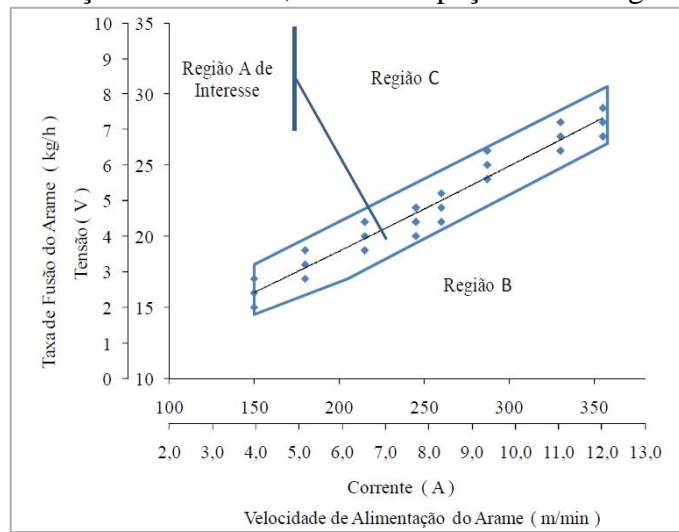


Fonte: Próprio autor

6.10 Teste do sistema no processo de soldagem MIG/MAG

Na Fig. 6.10.1 apresenta-se um envelope operacional obtido por Alves (2009), para ajustar os parâmetros certos no processo de soldagem MAG para juntas tipo T. Estes parâmetros serão utilizados para o teste do sistema no processo de soldagem das chapas metálicas.

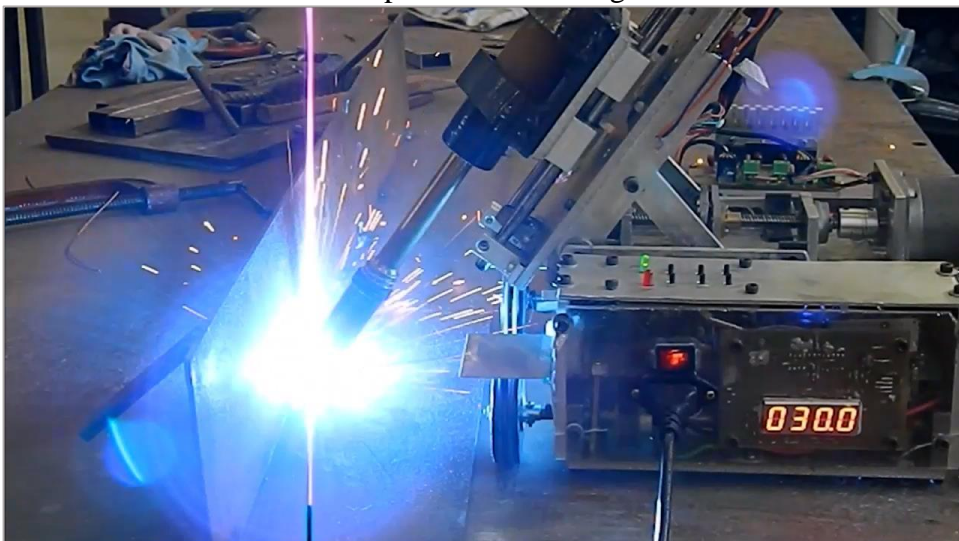
Figura 6.10.1 Envelope operacional para soldagem MAG robotizada (85% Ar + 15% CO₂), arame de aço ao carbono 1,2 mm com peças em configuração T



Fonte: (ALVES, 2009, p. 72)

Na Fig. 6.10.2, é apresentada uma imagem capturada do sistema num teste de soldagem.

Figura 6.10.2. Teste do sistema no processo de soldagem com um stickout de 12 mm.



Fonte: Próprio autor.

Capítulo 7. Análises de Resultados e Aportes Científicos

Por ainda parecer que o WLRA1.0 (assim chamado o robô) for um dispositivo funcional, pelos testes executados até agora, existem diversos aspectos da máquina que requerem ainda mais pesquisa e desenvolvimento para se tornar em um equipamento de qualidade comercial.

Das pesquisas feitas por diversos autores, por fim se entendeu que o processo de soldagem é sensível a variações de tensão-corrente, DBCP, velocidade de soldagem, soldabilidade do material, espessura das chapas, entre muitos outros fatores. Além disso que a fonte de soldagem MIG\MAG regula a corrente para manter a tensão constante, e que uma mudança na DBCP fará as demais variáveis agir o que afetará a transferência térmica, e conseqüentemente irá refletir numa má qualidade da peça terminada.

Segundo as pesquisas feitas o sistema automático que devia ser desenvolvido para o controle do posicionamento da tocha de soldagem no processo MIG\MAG, devia ser um sistema que mantivesse estável a posição do bico de contato e a velocidade do robô ao longo de processo de soldagem, garantindo influenciar minimamente nos parâmetros escolhidos. A corrente-tensão e a velocidade de alimentação de arame eram reguladas pela fonte de soldagem, pelo que somente havia possibilidade de mudar (ou regular) a DBCP e a velocidade da soldagem. Pelo que como produto final, como foi mostrado nas secções precedentes, foi desenvolvido um sistema mecatrônico que posiciona automaticamente a tocha e mantém essa posição estável ao longo do processo de soldagem. Para conseguir o posicionamento automatizado é medida a distância do robô com a peça vertical e são calculados os deslocamentos necessários dos mecanismos linear e angular, introduzindo o valor da distância medida nas equações geométricas obtidas na análise matemática. As equações geradas que descrevem geometricamente os deslocamentos dos mecanismos encontraram-se a partir das análises feitas nas Fig. 5.1.2, Fig. 5.2.35 e Fig. 5.2.36.

Foi criado um novo conceito para detecção das possíveis distorções de juntas em configuração T, baseado na medição de dois pontos que conformam uma geometria de linha reta, essa medição é feita utilizando dois sensores por ultrassom instalados estrategicamente no robô separados por uma distância de 207 mm, essa separação permite medir distorções a

partir de um milímetro entre os pontos que conformam a linha reta e assim corrigir a trajetória da soldagem.

Dentro dos intuitos da pesquisa estavam os projetos dos sistemas mecânicos, eletrônicos e os algoritmos de controle, com o principal interesse de adquirir uma visão integral no desenvolvimento de sistemas mecatrônicos. Esses projetos encontram-se nos Capítulos 4 e 5, onde foi detalhado por completo o dimensionamento dos subsistemas mecânicos e eletrônicos, melhorando o escopo no desenvolvimento deste tipo de sistemas, tendo que na maioria das pesquisas os sistemas mecatrônicos desenvolvidos utilizavam mecanismos existentes, ou bem os projetos mecânicos eram qualitativos sem grande justificativa de cálculo, situação similar com os sistemas eletrônicos, onde na maioria dos casos o chamado desenvolvimento mecatrônico era na realidade um desenvolvimento na programação ou nos aplicativos dos sistemas de software. Os testes feitos, e que foram apresentados no Capítulo 7 dos algoritmos de controle e posicionamento, foram satisfatórios, mas como foi dito anteriormente, ainda precisam-se muitas pesquisas e otimização dos mecanismos e subsistemas de controle criados.

Determinou-se que é fundamental a montagem dos sensores US-100, alinhados perfeitamente, isto é, que o ângulo de medição até a peça (refere-se à Fig. 5.1.2) seja o mesmo para que o cálculo da diferença não esteja polarizado pela detecção de outros objetos. Aconteceu em múltiplos testes que uma variação maior a 5 graus na janela de medição (refere-se à Fig. 5.1.3) produzia a detecção da tocha de soldagem e não da peça vertical, o que teve que ser corrigido muitas vezes.

A falta de robustez das peças mecânicas fabricadas pelas limitações do equipamento de precisão, produziam desalinhamentos na correção das trajetórias, pelo que tinha que se reajustar o mecanismo exaustivamente para o correto funcionamento. Um simples desajuste no sistema de alinhamento e balanço da direção fazia com que um dos servomotores não tivesse a força suficiente para dobrar a roda, isto como foi dito, produto das peças mecânicas fabricadas que tinham erros de fabricação superiores aos 2 mm (pela folga das máquinas mecânicas como o torno, a fresa e retífica). Mas acredita-se que a solução destes problemas não é colocar um servomotor de maior potência mecânica, pelo contrário é a fabricação das peças mecânicas utilizando um centro de usinagem automatizado baseado nos desenhos em 3D gerados no Solidworks®. O programa de Pós-graduação PPMEC, conta com um centro de usinagem HASS VF2 (DTS, HASS), para usinagem automatizada, utilizando a ferramenta

de software SolidCam® para geração do arquivo G-code, o que permitirá a automação na fabricação das peças mecânicas para o WLRA1.1. Embora esse centro de usinagem não estava em funcionamento no período desta pesquisa por falta de ferramentas, suporte de fresas e outros componentes, pelo que será uma sugestão a implantação do mecanismo futuro com ajuda desta máquina.

Os erros de posicionamento do sistema linear no processo de colocação da tocha na posição inicial da soldagem foram próximos aos 2mm (calculados na secção 6.9), contudo esse erro pode ser corrigido de duas maneiras:

1. Mudando o comprimento do eletrodo ou stickout, cujo valor no teste feito na secção 6.9, foi de 12mm, aumentando o stickout para 14mm, o erro no posicionamento será menor a 0,2mm. Lembrando que o comprimento do stickout depende do tipo de transferência metálica que para o caso da transferência por curto circuito o stickout pode tomar os valores entre 12mm e até 20mm.
2. Implementando a função de subir o descer a tocha de soldagem através dos botões na interface com o usuário, fazendo que cada vez que seja apertado o botão UP a tocha suba se afastando 1mm da peça. O caso contrário cada vez que seja apertado o botão DOWM a tocha desça 1mm em direção à peça.

Uma característica inédita do trabalho foi o projeto, desenho (vide Fig. 4.4.1), fabricação (vide Fig. 6.3.1) e validação do mecanismo angular (secção 6.7, 6.8 e 6.9) para o posicionamento e estabilização da tocha de soldagem. A estrutura e funcionalidade deste mecanismo comprovou-se ser apropriada para o posicionamento da tocha de soldagem em juntas tipo T, garantindo uma boa estabilidade na posição da tocha e na manutenção do arco elétrico no processo MIG/MAG.

Finalmente, cabe dizer que todos os objetivos propostos para o presente trabalho foram cumpridos aceitavelmente, tendo que foi projetado e implementado o sistema mecatrônico para o posicionamento da tocha e a manutenção do arco elétrico ao longo de um processo de soldagem em juntas em configuração T.

Capítulo 8. Conclusões.

De acordo com o trabalho realizado podemos concluir que:

1. É necessário manter estável a posição da tocha no processo de soldagem para estabilizar o arco elétrico e influenciar minimamente nos parâmetros de soldagem.
2. Os sensores de distância por infravermelho somente podem ser utilizados off-line no processo de soldagem, devido à alta intensidade luminosa do arco elétrico.
3. As frequências de emissão das ondas mecânico-acústicas da soldagem são muito baixas em comparação com as frequências de operação dos sensores de distância por ultrassom.
4. O abalancamento das cargas no mecanismo da direção utilizando o sistema de balanço e desalinhamento implementado, melhora em grande medida a operação dos servomotores.
5. O posicionamento da tocha de soldagem, baseado na medição da distância do carrinho com a peça vertical é uma técnica viável para automatizar esse processo e reduzir o tempo de treinamento do operador do equipamento.
6. O alinhamento dos sensores de ultrassom US-100 para a correção da trajetória de soldagem é necessário para o correto funcionamento do sistema completo.
7. O desenvolvimento de um sistema mecatrônico para soldagem robotizada é um processo de extrema complexidade que necessita de uma visão geral do processo de soldagem, do projeto mecânico, do projeto dos circuitos eletrônicos e da calibração e configuração de sensores e finalmente do domínio na programação de algoritmos computacionais.
8. Os motores de passo são dispositivos eletrônicos de alta precisão para aplicações de posicionamento e controle numérico.
9. O controle da velocidade de soldagem utilizando a técnica PWM e a medição com um encoder é uma alternativa de boa precisão ao serem utilizados motores de CC.
10. O motor de CC funciona perfeitamente para a movimentação do robô.
11. Os drivers de potência construídos para os motores de CC e de passo funcionaram perfeitamente.
12. Os objetivos propostos foram satisfeitos aceitavelmente, dado que conseguiu-se projetar e implementar o sistema mecatrônico para a manutenção do arco elétrico em juntas tipo T.

Capítulo 9. Sugestões para Trabalhos Futuros.

1. Otimização e construção de um sistema mecânico mais preciso, utilizando equipamento de alta precisão mecânica para tirar todas as folgas mecânicas.
2. Otimização do sistema de controle e unificação de toda a eletrônica, dado que os principais problemas no processo de teste dos algoritmos eram produto de falta de comunicação entre os subsistemas, e curto circuitos por contato.
3. Utilização de sistemas embarcados do tipo FPGA, para a solução da integração da eletrônica num sistema unificado.
4. Geração das funções de tecimento angular e linear.
5. Melhoramento dos algoritmos de automatização para simplificar o sistema.
6. Criação do sensor de seguimento de junta de soldagem (SSJ) sem tecimento baseado nas variáveis de arco elétrico.
7. Integração de sensores mais precisos para afinar os cálculos do sistema de controle.
8. Melhorar o sensor de velocidade da soldagem, utilizando um encoder com maior geração de pulsos por volta.
9. Integração de um sensor indutivo entre o carrinho e a peça horizontal para detecção do final da peça e evitar quedas do robô em altura.

Referências

ALEXANDER, K.CHARLES; SADIKU, N. O.MATTHEW. **Fundamentos de circuitos eléctricos**. 3ª Ed. Mc-Graw Hill, España, 2006, 880 p. Citado nas páginas 77, 102, 103 e 151.

ALPHA & OMEGA SEMICONDUCTOR. **AOD2N60**. Disponível em: <http://aosmd.com/pdfs/datasheet/AOD2N60.pdf> Data de acesso: 20/12/2014. Citado nas páginas 100, 103 e 107.

ALVARADO, J. PABLO. **Señales y Sistemas. Fundamentos Matemáticos**. 1ª Ed. Centro de Desarrollo de Material Bibliográfico, Cartago – Costa Rica, 2008, 337 p. Citado nas páginas 91, 95 e 96.

ALVES, VALTAIR J. **Desenvolvimento de envelopes operacionais para processo MIG/MAG robotizado com diferentes gases de proteção**. 2009. 172 folhas. Dissertação (Mestre em Engenharia) Universidade Federal de Rio Grande do Sul, Escola de Engenharia. Programa de Pós-Graduação em Engenharia Minas Metalúrgica e Materiais - PPGEM, Porto Alegre, Rio Grade do Sul - Brasil, 2009. Citado na páginas 22 e 160.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 7239: Chanfro de solda manual para construção naval**. Rio de Janeiro, 1982. 5 p. Citado na página 4.

BROERING, CARLOS E. **Desenvolvimento de sistemas para a automação da soldagem e corte térmico**. 2005. 127 pag. Dissertação (Mestrado em Engenharia Mecânica) – Programa de Pós-Graduação em Engenharia Mecânica – Universidade Federal de Santa Catarina, Santa Catarina, Brasil, 2005. Citado nas páginas 27 e 28.

CHAMBER, NICK. **Have you ever been on a cruise ship? Chances are it's robot-welded!** 25 de novembro de 2013, artigo online (Disponível em: <http://www.abb-conversations.com/2013/11/have-you-ever-been-on-a-cruise-ship-chances-are-its-robot-welded/>, Data de acesso, 14/12/2014) citado na página 3.

COSTA, AFRANIO R. **Desenvolvimento de um sistema de seguimento de junta que utiliza o próprio arco como sensor**. 2003. 120 pag. Dissertação (Mestrado em Engenharia Mecânica) – Programa de Pós-Graduação em Engenharia Mecânica – Universidade Federal de Santa Catarina, Santa Catarina, Brasil, 2003. Citado nas páginas 12, 13 e 16.

E-GIZMO. **Ultrasonic Sonar Module - US-100**. Disponível em: <http://www.e-gizmo.com/KIT/images/ultrasonicsonar/ultrasonic%20sonar%20module%20r0.pdf>. Data de acesso: 20/12/2014. Citado nas páginas 63, 64, 68, e 167.

ERDMAN, G.ARTHUR; SANDOR, N. GEORGE. **Diseño de mecanismos: Análisis y Síntesis**. 3ª Ed. Mexico: Prentice Hall, 1998. p. 664. Citado nas páginas 39, 41, 49, 50, 56, 57 e 58.

ESAB, **Tocha PMC400-2K**, Apostila. Data de acesso: 14/10/2013. Disponível em: http://www3.esab.com.br/literatura/pdf/0206642_rev3_PistolaPMC400-2K_pt.pdf. Citado na página 39.

ESAB, **Equipamentos de soldagem**. Apostila. Disponível em: http://www.esab.com.br/br/por/Produtos/equipamentos/soldagem_mig/upload/1901888_rev-2_OrigoFeed-302N-P0_304N-P4_Pt.pdf, Data de acesso: 14/10/2013. Citado na página 39.

FATHI, UZUN; NEZIHI, B. ALI. **Immersion ultrasonic technique for investigation of total welding residual stress**. Procedia Engineering, v. 10, p. 3098–3013, 2011. Citado na página 28.

FERDINAND, P. B.; E. RUSSELL, JOHNSTON JR. **Mecânica Vetorial para Engenheiros: Estática**. 5ª Ed. Rev. São Paulo: Makron Books, 1994. 793 pag. Não citado.

FERREIRA S., CARLOS E. **Desenvolvimento de um sistema Mecatrônico para soldagem multipasses monitorado por visão computadorizada**. 1997 . 165 pag. Dissertação (Mestrado em Engenharia) – Centro Técnico Aeroespacial - Instituto Tecnológico de Aeronáutica. Área de Pós-Graduação em Mecatrônica e Dinâmica de Sistemas, São Paulo, Brasil, 1997. Citado nas páginas 9, 10 e 11.

GOHR J., RAUL. **Novos métodos de controle para soldagem MIG/MAG**. 2002. 165 pag. Teses (Doutorado em Engenharia Mecânica) – LAPROSOLDA – Universidade Federal de Santa Catarina, Santa Catarina, Brasil, 2002. Citado nas páginas 11 e 12.

HASS AUTOMATION.VF-2. Disponível em: http://www.haascnc.com/mt_spec1.asp?id=VF-2&webID=40_TAPER_STD_VMC#gsc.tab=0 Data de acesso: 20/12/2014. Citado na página 167.

INTERIANO, EDUARDO. **Sistemas em tempo discreto**. Apostila. Data de acesso: 24/8/2014. Disponível em: <http://www.ie.itcr.ac.cr/einteriano/control/index.php>. Citado nas páginas 83, 84 e 91.

KALIL, M. FELIPE. **Evolução da técnica de seguimento de junta via sensoriamento do arco para operações de soldagem em posições forçadas**. 2013. 136 pag. Dissertação (Mestrado em Engenharia Mecânica) – Programa de Pós-Graduação em Engenharia Mecânica – Universidade Federal de Santa Catarina, Santa Catarina, Brasil, 2009. Citado na página 16, 17 e 23.

KIM, ILL-SO et al. **A study on the quality improvement of robotic GMA welding process**. Journal of Robotics and Computer-Integrated Manufacturing, v. 19, p. 567–572, 2003. Citado na página 12.

KOU; SINDO. **Modeling, Sensing and Control of Gas Metal Arc Welding**. New Jersey - USA, John Wiley & Sons, p. 196, 2003. Citado na página 24, 25, e 62.

KUO, C. BENJAMIN. **Sistemas de Control Automático**. 7ª Ed. Prentice Hall Hispanoamérica, México, 1996, 931 p. Citado na pagina 61, 88, 89, 92, 95 e 149.

LEE, D. et al. **Development of a mobile robotic system for working in the double-hulled structure of a ship.** Journal of Robotics and Computer-Integrated Manufacturing, v. 26, p. 13–23, 2010. Citado na página 14.

LEE, D. et al. **Development and application of an intelligent welding robot system for shipbuilding.** Journal of Robotics and Computer-Integrated Manufacturing, v. 27, n. 2, p. 377–388, Abr. 2011. Citado nas páginas 15 e 18.

LINCOLN ELECTRIC. **The Welding Experts. Welding distortion.** (Disponível em: <http://www.lincolnelectric.com/en-gb/support/welding-how-to/Pages/welddistortion-detail.aspx> Data de acesso: 22 de outubro de 2013). Citado na página 26.

MACHADO, MARCUS V. R. **Sistema embarcado sem fio para monitoramento de sinais na soldagem a arco elétrico com abordagem tecnológica.** 2011. 127 folhas. Dissertação (Mestrado em Engenharia Mecânica) Universidade Federal de Uberlândia, Faculdade de Engenharia Mecânica. Programa de Pós-Graduação em Engenharia Mecânica, Uberlândia, Minas Gerais, Brasil, 2011. Citado na página 138.

MAXIM. **Serially Interfaced, 8 bit LED Display Driver MAX7219.** Disponível em: <http://datasheet.octopart.com/MAX7219CWG-Maxim-datasheet-98628.pdf> Data de acesso: 20/12/2014. Citado nas páginas 135 e 136.

MICROCHIP. **PIC16F877A Datasheet.** Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf> Data de acesso: 20/12/2014. Citado nas páginas 69, 100 e 150.

MICROCHIP. **PIC18F4550 Datasheet.** Disponível em: <http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf> Data de acesso: 20/12/2014. Citado na páginas 121 e 128.

MICROCHIP. **PIC10F322 Datasheet.** Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/40001585B.pdf> Data de acesso: 20/12/2014. Citado na página 72 e 73.

MINEBEA. **23LM-XXX Bipolar Stepper Motor.** Disponível em: http://cnc25.free.fr/documentation/moteurs%20pap/pap_nmb.pdf Data de acesso: 20/12/2014. Citado nas páginas 124, 126 e 127.

MODENESI, J. PABLO. **Introdução à Física do Arco Elétrico.** Belo Horizonte - Brasil, 2011a. Disponível em: <<http://www.iem.unifei.edu.br/professores/edmilson/Arco.pdf>>. Data de acesso: 24/03/2014. Citado nas páginas 20, 21 e 22.

MODENESI, J. PABLO. Soldagem I: **Introdução aos Processos de Soldagem.** Belo Horizonte - MG, Brasil, 2011b. Citado nas páginas 19, 37, 61 e 63 .

MODENESI, J. PABLO et. al. Soldagem I: **Introdução à Metalurgia da Soldagem.** Belo Horizonte - MG, Brasil, 2011c. Citado na página 61.

NAIDU; D.S. **Modeling, Sensing and Control of Gas Metal Arc Welding.** USA, Elsevier, p. 372, Junho 2003. Citado nas páginas 27 e 28.

NASCIMENTO, S. D. ALEXANDRE. **Fenomenologia da soldagem MIG/MAG-PV e sua aplicabilidade para juntas sobreposta e em V (passe de raiz)**. 2011. 186 folhas. Tese (Doutorado em Engenharia Mecânica) – Universidade Federal de Uberlândia, Escola de Engenharia. Programa de Pós-Graduação em Engenharia Mecânica, Uberlândia, Minas Gerais, Brasil, 2011. Citado na página 22.

NSK. **Ball Bearing Catalog**. Disponível em: http://www.jp.nsk.com/app01/en/ctrq/index.cgi?gr=dn&pno=NSK_CAT_E1102m_B46-75. Data de acesso 4/03/2013. Citado na página 55.

ON SEMICONDUCTOR. **2N4401: General Purpose Transistor**. Disponível em: <http://www.onsemi.com/pub/Collateral/2N4401-D.PDF> Data de acesso: 20/12/2014. Citado na página 102.

TOR. **2N3904: General Purpose Transistor**. Disponível em: <http://www.onsemi.com/pub/Collateral/2N3903-D.PDF> Data de acesso: 20/12/2014. Citado na página 102.

PAN, ZHENGXI. et al. **Recent progress on programming methods for industrial robots**. Journal of Robotics and Computer-Integrated Manufacturing, v. 28, p. 87-94, 2012, Citado na página 17.

PICHARDO, C. JESÚS. **Compensación de Zona Muerta y Variación de Carga en el Control de Velocidad de Motores de Corriente Directa**. Querétaro- México. 2008. Dissertação de Mestrado em Ciências de Tecnologia Avançada, Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, Instituto Politécnico Nacional, Queretaro-Qro. 112 p. Citado nas páginas 77 e 87.

PIMENTA, M. CAROLINA. **Sistema de visão por infravermelho próximo para monitoramento de processos de soldagem a arco**. 2011. 151 folhas. Dissertação (Mestrado em Engenharia) - Universidade Federal de Uberlândia, Escola de Engenharia. Programa de Pós-Graduação em Engenharia Mecânica, Uberlândia, Minas Gerais, Brasil, 2011. Citado nas páginas 29 e 62.

POLEZI, RAFAEL L, et al. **Sistema de geração de trajetórias em manipulador cartesiano para aplicações de soldagem naval**. CONEM 2012, v. 7, Maranhão - Brasil, p. 10, Jul.-Ago. 2012. Citado na página 16.

PRESMAN, ABRAHAM I. et al. **Switching Power Supply Design**. 3ª Ed. Mc-Graw Hill, USA, 2009, 880 p. Citado na página 103.

PRITSCHOW, G. **Automatic Programing of Industrial Robot by Sensor Guidance**. Journal of Robotics and Computer-Integrated Manufacturing, v. 5, n. 2/3, p. 173–181, 1989. Citado nas páginas 7, 8 e 11.

QUIJANO, O. G. MANUEL; HERNANDEZ, C. G. CARLOS. **Obtención experimental de los parámetros del motor que se utilizará en el sistema de locomoción de una esfera rodante**. Bucaramanga - Colombia. 2009. Trabajo de Grado para obter por el título de Especialista en Control e Instrumentación Industrial. Universidad Pontificia Bolivariana.

Escuela de Ingeniería y Administración, Facultad de Ingeniería Electrónica, Bucaramanga, 48 p. Citado na página 77.

RASHID, H. MUHAMMAD. **Electrónica de Potencia: circuitos, dispositivos y aplicaciones**. 2ª Ed. Prentice Hall Hispanoamérica, México, 1993, 720 p. Citado na página 102.

RADAJ; DIETER. **Welding Residual Stresses and Distortion: Calculations and Measurement**. Vol. 2, USA, Woodhead Publishing, Limited, p. 397, 2003. Citado na página 25.

RichardsonFPD, **High Frequency (HF) Welding Inverter**, Disponível em http://www.richardsonrfpd.com/resources/RelDocuments/SYS_28/HF_Welding_Inverte_r.pdf, Data de acesso 22 de junho de 2013. Citado na página 31.

SANDLER, B. Z. **Robotics: Designing the Mechanisms for Automated Machinery**. 2ª Ed. USA: Academic Press, 1999. p. 444. Citado na página 39.

SEDRA; SMITH. **Microelectronics Circuits**. 5. ed. USA: Oxford University Press, 2006. Citado nas páginas 34 e 102.

SHARP.GP2Y0A21YK. Disponível em: http://www.sharpsma.com/webfm_send/1489
Data de acesso: 20/12/2014. Citado nas páginas 117 e 118.

SHIGLEY, E. JOSEPH. **Projeto de engenharia mecânica**. 7ª Ed. Porto Alegre – Brasil, Bookman, 2005, p. 994. Citado nas páginas 43, 44, 45, 47, 48 e 55.

SHIGLEY, E. JOSEPH. **Mechanical engineering design**. 8ª Ed. USA, Mc Graw Hill, 2006, p. 1514. Citado nas páginas 56, 57, 61, 62, 63, 64 e 67.

SLOCUM, H. ALEXANDRE. **An automated shear stud welding system**. Journal of Robotics and Autonomous Systems, v. 6, p. 387–382, 1990. Citado nas páginas 47, 53, 54, 55, 58 e 59.

ST. **L6203 CMOS Full Bridge Driver**. Disponível em: http://www.st.com/web/catalog/sense_power/FM142/CL851/SC1790/SS1556/PF63230?sc=in_ternet/analog/product/63230.jsp Data de acesso: 20/12/2014. Citado nas páginas 122, 123, 124 e 127.

STEINBACH, CARVALHO R. et al. **Implementação de controlador CNC de baixo custo em manipulador robótico para soldagem**. Congresso Nacional de Engenharia Mecânica, v. 5, p. 9, 2008. Citado nas páginas 13 e 14.

STEINBACH, CARVALHO R. **Robô CNC a automação da soldagem MIG/MAG em posições e situações de extrema dificuldade**. 2009. 136 pag. Dissertação (Mestrado em Engenharia Mecânica) – Programa de Pós-Graduação em Engenharia Mecânica – Universidade Federal de Santa Catarina, Santa Catarina, Brasil, 2009. Citado na página 14.

TOCCI, RONALD; WIDMER, S. NEAL. **Sistemas digitais, principios e aplicaciones**. 8ª Ed. Pearson Educación, México, 2003, 912 p. Citado nas páginas 131, 132 e 133.

TT Electronics, *Welding Power Supplies*, Disponível em: <http://www.bitechnologies.com/pdfs/weldingpowersupplies-an.pdf>, Data de acesso 22 de junho 2013. Citado na página 30.

VILSON J. LEFFA. *Normas da ABNT, Citações e Referências Bibliográficas* (Disponível em: <http://www.leffa.pro.br/textos/abnt.htm#5.16>. Data de acesso: 19 de abril de 2013). Não citado.

WILDI, THEODORE. *Máquinas eléctricas y Sistemas de Potencia*. 6^a Ed. Pearson Educación, México, 2007, 960 p. Citado na página 128.

WEMAN; KLAS. *Welding processes handbook*. FL - USA, CRC Press, p. 468, 2003. Citado na página 4.

APÊNDICE A

A.1 Código do Microcontrolador PIC16F877A para o Controle da Velocidade e Correção da Trajetória de Soldagem.

ARQUIVO MAIN.C

```

#include "C:\Users\Leonardo\SkyDrive\Documentos\main.h"
#include "main.h"
//#include "Torch.h"
#include "IO_main.h"
//#include <float.h>
#include <math.h>
//#include <stdlib.h>

//////////////////////////////////Parametros do 7 segmentos//////////////////////////////////

//////////////////////////////////
//          CONSTANTES CONFIGURACAO DO MAX 7219
#define COMENZAR    0x09 //Señal START
#define FINALIZAR   0x05 //Señal STOP
#define RESETEAR    0xFF
#define UP_MSG      0x22
#define DOWN_MSG    0x11
//#define DUTY_100 530 // PWM 24V partiendo de 24V ----> 100%
//#define DUTY_MIN 86 // PWM 1,06V partiendo de 24V ----> 16,6%
#define DUTY_MAX 362 // PWM 12V partiendo de 24V ----> 68,3.13%
//#define FREQ_PWM 30 // KHz
#define ZERO 0x0000
#define VEL_MIN 0x0150

//////////////////////////////////Constantes do sistema//////////////////////////////////
//int DADOS_DIG[16] = {Z,U,D,T,Q,C,S,SET,O,N,R,E,H,EL,P,B};
int AD_DIG[8] = {0x08,0x06,0x06,0x05,0x04,0x03,0x02,0x01};
int DADOS_INI [5] = {0x01,0x00,0xFF,0x0F,0x07};
int AD_INI[5] = {0x0C,0x0F,0x09,0x0A,0x0B};

int16 Real_Speed = 0x0150;
int16 Velocidad_Final = 150;
int16 PWM_SET_POINT = 0;
int8 count_ADC = 0;
int16 Valor = 0;
int16 c = 0;
int1 VEL_INI = FALSE;
int16 DL = 0x0000; // True value of left distance sensor
int16 DR = 0x0000; // True value of right distance sensor
signed int16 DIR = 0x0000; // Left or right data
signed int16 ek = 0;
signed int16 erro = 0;
signed int32 erro_c = 0;
int16 mk = 0;
//int16 ek1 = 0;
int16 Ciclo_PWM = 95;

```

```

int32 counter = 0; // Recibe o valor do RTCC
int16 Valor_atual = 0;
int32 vel_medida = 0; // velocidad calculada no modulo de velocidade
boolean CTR_P = FALSE; // Variable de control para detección de pulso
int1 Write_Slave = FALSE;
int1 UPDATE = 0;
int1 COMP = 0;
int16 delta_c = 0;
int8 ctr = 0;
int i = 0;
int16 count_led = 0;
int estado = 0;
int8 timer0 = 0;
boolean parar = TRUE;
int8 pulsos_p = 0;
int8 pulsos_n = 0;
//int1 pulso = 0;
boolean APAGADO = FALSE;

////////// D3 D2 D1 D0
////////// 1001 1001 1001 1001
#include "Functions.h"

#INT_TIMER0
void rtcc_isr(){
    c = c + 1;
}

//////////FUNCAO MAIN DO PROGRAMA //////////
//////////
void main()
{

    setup_adc_ports(AN0);
    setup_adc(ADC_CLOCK_DIV_32); // TAD = 2us
    setup_psp(PSP_DISABLED);
    setup_spi(FALSE);
    setup_timer_0(RTCC_EXT_L_TO_H|RTCC_DIV_1);// tempo do timer_0 20 ms por conta
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_4);// tempo do timer 1 us por conta
    setup_timer_2(T2_DIV_BY_1,132,1); //setup_timer_2(T2_DIV_BY_1,132,1);
    setup_ccp1(CCP_PWM);
    SET_PWM1_DUTY(0); //DUTY CYCLE DO PWM1
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);

    // Establece los pines como entrada o salida: 0 es salida y 1 es entrada

    Set_tris_A(TRISA); // TRISA
    Set_tris_B(TRISB); // TRISB
    Set_tris_C(TRISC); // TRISC I2C INPUT
    Set_tris_D(TRISD); // TRISD
    Set_tris_E(TRISE); // TRISE

    output_low(SDO);
    output_high(CARGA);

```

```

output_high(EN_OBS);

output_low(TRIG_D);
output_low(TRIG_I);

delay_ms(20);

output_high(RS);
output_high(LS);

APAGADO = In_Segmentos(0,APAGADO);

WRITE(RESETEAR);
delay_ms(500);

do {

count_led = LED_BLIND(count_led, estado); // estado_5 --> 6 us

////////////////////////////////////Maquina de estados principal //////////////////////////////////////

switch(estado) // 6 us
{
case 0: //////////////////////////////////////ESTADO ZERO

if(!input(PROG) && (ctr == 0)){
set_rtcc(0);
ctr = 1;
timer0 = 0;
}else if (input(PROG)){
timer0 = 0;
ctr = 0;
}
else if ((ctr > 0) && (timer0 > 15) && !input(PROG))ctr = 2;
else timer0 = get_rtcc();

//////////////////////////////////// AJUSTE DO SERVO DE DIRECAO //////////////////////////////////////

// Filtragem do sistema de medição de distância
DR = 0;
DL = 0;

for (i = 0; i<4; i++){
DR = (DR + distância(1)); // distância(mm) = DR * 0,167 VAR == 1
DL = (DL + distância(0)); // distância(mm) = DL * 0,167 VAR == 0
}
DR = DR/4;
DL = DL/4;

if(!input(RST))
{
disable_interrupts(INT_TIMER0);
disable_interrupts(GLOBAL);
setup_timer_0(RTCC_EXT_L_TO_H|RTCC_DIV_1);
output_low(TRIG_D);
output_low(TRIG_I);
}
}

```

```

set_rtcc(0);
count_led = 0;
i = 0;
WRITE(RESETEAR);
ctr = 0;
Real_Speed = 0x0150;
Velocidad_Final = 150;
timer0 = 0;
c = 0;
APAGADO = FALSE;
APAGADO = In_Segmentos(0,APAGADO);
output_high(EN_OBS);
COMP = 0;
estado = 0;
vel_medida = 0;
counter = 0;
DR = 0;
DL = 0;
count_ADC = 1;
Valor = 0;
Write_Slave = FALSE;
parar = TRUE;
SET_PWM1_DUTY(0); //DUTY CYCLE DO PWM1
WRITE(PARAR);
delay_ms(100);
WRITE(RESETEAR);

} else if (!input(PROG) && (ctr == 2) && (DR < 1220) && (DR > 830)) {
estado = 1;
Segmentos(VEL_MIN,0);
Segmentos(ZERO,1);
timer0 = 0;
c = 0;
ctr = 0;
} else if((DL < 1220) && (DR < 1220) && (DR > 830) && (DL > 830) && !APAGADO){

Segmentos(BIN_BCD(DR),0); // Muestra la distância medida en el sensor del frente
Segmentos(BIN_BCD(DL),1); // Muestra la distância medida en el sensor de atrás

estado = 0;
} else {

APAGADO = In_Segmentos(1,APAGADO);
output_high(RS);
output_high(LS);
delay_ms(150);
estado = 0;
}

break;

case 1: //***** ESTADO SET POINT DE VELOCIDAD*****

Segmentos(estado,1);

if(!input(UP) | !input(DOWN) | !input(PROG)) && (ctr == 0)){

```

```

set_rtcc(0);
ctr = 1;
timer0 = 0;
} else if (input(UP)&&input(DOWN)&&input(PROG)){
ctr = 0;
timer0 = 0;
c = 0;
}
else if ((timer0 > 7) && (timer0 < 63) && (ctr == 1) && (c == 0)){
ctr = 2;
}
else if ((timer0 >= 63) && (timer0 < 204) && (ctr == 2) && (c == 0)){
ctr = 3;
}
else if ((timer0 >= 204) && (ctr == 3)){
c = c + 1;
ctr = 4;
set_rtcc(0);
timer0 = 0;
} else timer0 = get_rtcc();

if(!input(RST)) estado = 0;
else if ((ctr > 1) && !input(UP)){ estado = 2;
//Segmentos(BIN_BCD(Ciclo_PWM + 1),0);
Segmentos(Real_Speed,0);
Segmentos(ZERO,1);
}
else if ((ctr > 1) && !input(DOWN)){ estado = 3;
//Segmentos(BIN_BCD(Ciclo_PWM - 1),0);
Segmentos(Real_Speed,0);
Segmentos(ZERO,1);
}
else if ((ctr == 3) && !input(PROG)) {estado = 4;
//Segmentos(BIN_BCD(Ciclo_PWM),0);
Segmentos(Real_Speed,0);
Segmentos(ZERO,1);
Real_Speed = BIN_BCD(Velocidad_Final);
PWM_SET_POINT = PWM_CT_Calc(Velocidad_Final); // Ciclo de trabajo inicial
c = 0;
ctr = 0;
}
else estado = 1;
//////////////////// Letura dos sensores de distância////////////////////
break;

//DUTY CYCLE DO PWM1

//*****ESTADO UM TERMINA AQUI*****//

case 2: //////////////////////ESTADO INCREMENTO DO SET POINT DE VELOCIDADE

Segmentos(estado,1);

if (ctr==2){
// acrescenta numa unidade a velocidade do carrinho

```

```

// Identifier = 0 produz operacao Suma + 1
Velocidad_Final = Operacion(Velocidad_Final,0);
Real_Speed = BIN_BCD(Velocidad_Final);
Segmentos(Real_Speed,0);
delay_ms(300);

} else if (ctr==3) {

// Identifier = 0 produz operacao Suma + 1 rapidamente

Velocidad_Final = Operacion(Velocidad_Final,0);
Real_Speed = BIN_BCD(Velocidad_Final);
Segmentos(Real_Speed,0);
delay_ms(150);
}

else if(ctr==4){
// acrescenta em 10 unidades a velocidade do carrinho

// identifier = 1 produz Suma + 10

Velocidad_Final = Operacion(Velocidad_Final,1);
Real_Speed = BIN_BCD(Velocidad_Final);
Segmentos(Real_Speed,0);
delay_ms(500);
}

estado = 1;
break; // case 2 do switch estado

case 3: //////////// ESTADO DIMINUIÇÃO DO SET POINT DE VELOCIDADE

//Segmentos(estado,1);

if(ctr == 2){

// identifier = 2 produz Operacion Resta - 1

Velocidad_Final = Operacion(Velocidad_Final,2);
Real_Speed = BIN_BCD(Velocidad_Final);
Segmentos(Real_Speed,0);
delay_ms(300);

} else if (ctr == 3){

// identifier = 2 produz Operacion Resta - 1

Velocidad_Final = Operacion(Velocidad_Final,2);
Real_Speed = BIN_BCD(Velocidad_Final);
Segmentos(Real_Speed,0);
delay_ms(150);

} else if(ctr==4){
// acrescenta em 10 unidades a velocidade do carrinho
// identifier = 3 produz Resta - 10

```



```

Velocidad_Final = Operacion(Velocidad_Final,3);
Real_Speed = BIN_BCD(Velocidad_Final);
Segmentos(Real_Speed,0);
delay_ms(500);

}
estado = 1;

break; // case do switch estado 3

case 4: ////////////////////////////////// ESTADO DE ESPERA DO INICIO DA SOLDAGEM

if(!Write_Slave) {
WRITE(COMENZAR);
Write_Slave = TRUE;
}

if((!input(START)|!input(UP))|!input(DOWN) && (ctr == 0)){
set_rtcc(0);
ctr = 1;
timer0 = 0;
} else if (input(START)&&input(UP)&&input(DOWN)){
ctr = 0;
timer0 = 0;
}
else if (!input(START) && (timer0 > 63) && (ctr == 1))ctr = 3;
else if ((!input(UP) || !input(DOWN)) && (timer0 > 7) && (ctr == 1)) ctr = 2;
else timer0 = get_rtcc();

//Segmentos(BIN_BCD(Ciclo_PWM),0);
set_adc_channel(0);
output_low(EN_OBS);

DR = 0;
DL = 0;

for (i = 0; i<4; i++){
DR = (DR + distância(1)); // distância(mm) = DR * 0,167 VAR == 1 ---- OPT DR + 25 -- MAX 1225us
DL = (DL + distância(0)); // distância(mm) = DL * 0,167 VAR == 0 ---- OPT DR + 25 -- MAX 1225us
}

DR = DR/4;
DL = DL/4;

SERVO_DIRECAO(DR,DL);

if(!input(RST)) estado = 0;
else if ((ctr == 3) && !input(START)) {
estado = 5;

setup_timer_0(RTCC_INTERNAL|RTCC_DIV_256);
enable_interrupts(INT_TIMER0);
enable_interrupts(GLOBAL);

```

```

Segmentos(Real_Speed,0);
Ciclo_PWM = (PWM_SET_POINT);
SET_PWM1_DUTY(Ciclo_PWM);
CTR_P = FINDING_FIRST_PULSE(Ciclo_PWM); //

if(!CTR_P) UPDATE = 0;
else UPDATE = 1;
c = 0;

//Valor = read_adc();
//Valor_atual = Valor;

}
else if ((ctr == 2) && !input(UP)){ WRITE(UP_MSG);
ctr = 0;
delay_ms(1000);
}
else if ((ctr == 2) && !input(DOWN)){ WRITE(DOWN_MSG);
ctr = 0;
delay_ms(1000);
}
else estado = 4;

break;

case 5: ////////////////////////////////// ESTADO PROCESO DE SOLDAGEM

/*
if((count_ADC == 127) && VEL_INI){ // 3us

delta_c = Valor - Valor_atual; // 3 us
erro_c = (1000*(delta_c))/Valor_atual; // 13us
Valor_atual = Valor_atual + 9*(delta_c)/100; // 10us
Ciclo_PWM = Current_Control (erro_c,Ciclo_PWM); //38us
SET_PWM1_DUTY(PWM_CT_Calc(Ciclo_PWM)); // 32us
count_ADC = count_ADC + 1; // 1us
}else if ((count_ADC == 128) && VEL_INI) count_ADC = 0; // 2us
//else if(VEL_INI) count_ADC = count_ADC + 1; // 2us

Valor = Current_measurement(Valor,count_ADC); // 66us
*/

/////////////////////////////////Letura da velocidade //////////////////////////////////

if((pulsos_p == 10) && UPDATE){ // MEDIR PULSO BAJO 20us

//erro = (1000*(Velocidad_Final - vel_medida))/Velocidad_Final;
//Ciclo_PWM = Ciclo_PWM + (erro*Ciclo_PWM)/1000;
Segmentos(BIN_BCD(Vel_medida),0); // 206 us
erro = PWM_SET_POINT - PWM_CT_Calc(vel_medida); // 270us
Ciclo_PWM = Ciclo_PWM + erro/2; // 20us
SET_PWM1_DUTY(Ciclo_PWM); // 100us
//Valor_atual = Valor; // 10us
pulsos_p = 0; // 1us

```

```

} else if ((pulsos_n == 10) && !UPDATE){ //MEDIR PULSO ALTO 20us

    //erro = (1000*(Velocidad_Final - vel_medida))/Velocidad_Final;
    //Ciclo_PWM = Ciclo_PWM + (erro*Ciclo_PWM)/1000;
    Segmentos(BIN_BCD(vel_medida),0);
    erro = PWM_SET_POINT - PWM_CT_Calc(vel_medida);
    Ciclo_PWM = Ciclo_PWM + erro/2;
    SET_PWM1_DUTY(Ciclo_PWM);
    //Valor_atual = Valor;
    pulsos_n = 0;
}

//////////Deteccion de PULSOS

if(input(PULSE_B) && !CTR_P){ // 2us
delay_ms(24); // 24ms
if(input(PULSE_B)){ // 10us
pulsos_p = pulsos_p + 1; // 5us
CTR_P = 1; // 5us
//output_high(LED_PROG); // 20us
//output_low(LED_START); // 20us
if(UPDATE){ // 5us
vel_medida = SPEED_MEASUREMENT(VEL_INI,vel_medida,Velocidad_Final,c);// 480us
VEL_INI = 1; // 5us
c = 0; // 1us
}
SERVO_DIR(DR,DL); // (DL,DR) para reversa y (DR,DL) para adelante // 30us
}
}else if (!input(PULSE_B) && CTR_P){ // 20us
delay_ms(24); // 24ms
if(!input(PULSE_B)){ // 10us
pulsos_n = pulsos_n + 1; // 5us
CTR_P = 0; // 5us
//output_low(LED_PROG); // 20us
//output_high(LED_START); // 20us
if(!UPDATE){ // 5us
vel_medida = SPEED_MEASUREMENT(VEL_INI,vel_medida,Velocidad_Final,c); // 480us
VEL_INI = 1; // 5us
c = 0; // 1us
}
SERVO_DIR(DR,DL); // (DL,DR) para reversa y (DR,DL) para adelante // 30 us
}
}

DR = (DR + distância(1))/2; ///Distance Sensor Front // 30us /// Wait for 1/2 Timer0 count
// 32us
DL = (DL + distância(0))/2; ///Distance Sensor Back // 30us /// Wait for 1/2 Timer0 count
// 32us

//SERVO_DIR(DR,DL);

if(!input(RST)) estado = 0; // 15us
else if (!input(STOP) || (DR > 1256) || !input(IN_OBS)) estado = 6; // 60us
else estado = 5; // 10us

```


ARQUIVO MAIN.H

```
#include <16F877A.h>
```

```
#device adc=10
```

```
#FUSES NOWDT           //No Watch Dog Timer
```

```
#FUSES HS              //High speed Osc (> 4mhz)
```

```
#FUSES NOPUT          //Power Up Timer
```

```
#FUSES PROTECT        //Code protected from reads
```

```
#FUSES NODEBUG        //No Debug mode for ICD
```

```
#FUSES NOBROWNOUT    //No brownout reset
```

```
#FUSES NOLVP          //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
```

```
#FUSES NOCPD          //Data EEPROM Code Protected
```

```
#FUSES WRT_50%        //Lower half of Program Memory is Write Protect
```

```
#use delay(clock=16000000)
```

ARQUIVO IO_MAIN.H

```
////////////////////////////////PORT A////////////////////////////////
```

```
#define SENSE Pin_A0 // 1 Input current measurement => PIN 2
//#define NC Pin_A1 // 1 System Reset => PIN 3
#define RST Pin_A2 // 1 System Reset => PIN 4
//#define NC Pin_A3 // 1 => PIN 5
#define CLK_TO Pin_A4 // 1 CLK -> 50.0 Hz or 19.8ms PIN 6
#define NOUSAR Pin_A5 // 1 Output signal ENABLE for L6203 => PIN 7
//A6 1
//A7 1 ->>>> PORT 1101 1111
#define TRISA 0xFF
```

```
//////////////////////////////// PORT B //////////////////////////////////
```

```
// Las entradas Q son activas en bajo su estado natural es alto
#define PULSE_B Pin_B0 // 1 Input Sensor Velocidade (Encoder)=> PIN 33
#define PULSE_A Pin_B1 // 1 Input Sensor Velocidade (Encoder)=> PIN 34
#define TRIG_D Pin_B2 // 0 Output Distance Trigger Right => PIN 35
#define ECHO_D Pin_B3 // 1 Input Distance Trigger Right => PIN 36
#define ECHO_I Pin_B4 // 1 Input Distance Trigger Left => PIN 37
#define TRIG_I Pin_B5 // 0 Output Distance Trigger Left => PIN 38
#define EN_OBS Pin_B6 // 0 Output ENABLE IR sensor Switch => PIN 39
#define IN_OBS Pin_B7 // 1 Input IR Sensor Switch => PIN 40
// TRISB ->>>> 1001 1011
#define TRISB 0x9B
```

```
////////////////////////////////PORT C////////////////////////////////
```

```
//Las salidas son activas en alto su estado natural es bajo
#define RS Pin_C0 // 0 Output to SERVO (TURN RIGHT) => PIN 15
#define LS Pin_C1 // 0 Output to SERVO (TURN LEFT) => PIN 16
#define EN_PWM Pin_C2 // 0 PWM1 OUTPUT TO ENABLE L6203 => PIN 17
#define CARGA Pin_C3 // 0 LOAD SIGNAL SERIAL => PIN 18
#define SDO Pin_C4 // 0 SDA I2C SERIAL DATA SIGNAL => PIN 23
//#define NC Pin_C5 // 1 => PIN 24
#define LED_PROG Pin_C6 // 0 Programming Led signal output => PIN 25
#define LED_START Pin_C7 // 0 START Led signal output => PIN 26
//
// TRISC ->>>> 0010 0100
#define TRISC 0x20 // Defined for I2C as INPUT
```

```
////////////////////////////////PORT C////////////////////////////////
```

```
//Las salidas son activas en alto su estado natural es bajo
```

```
#define DOWN Pin_D0 // 1 Bottom down speed => PIN 19
#define PROG Pin_D1 // 1 Bottom programming speed => PIN 20
#define START Pin_D2 // 1 Bottom start welding => PIN 21
#define STOP Pin_D3 // 1 Bottom stop welding => PIN 22
#define UP Pin_D4 // 1 Bottom up speed => PIN 27
#define LOAD Pin_D5 // 0 CLK signal to MAX7219 => PIN 28
#define DIN Pin_D6 // 0 Load output to MAX7219 => PIN 29
#define CLK_7S Pin_D7 // 0 Serial data output to MAX721 => PIN 30
// TRISD ->>>> 0001 1111
#define TRISD 0x1F
```

```
////////////////////////////////PORT E //////////////////////////////////
#define CCP1 Pin_E2 // 1 INPUT PWM Signal => PIN 10
#define FIM_CM1 Pin_E1 // 0 OUTPUT Signal Direction for L6203 => PIN 9
#define FIM_CM2 Pin_E0 // 0 OUTPUT Signal Direction for L6203 => PIN 8

// TRISE ->>> 1111 1100
#define TRISE 0xFF

////////////////////////////////PORT ADDRESS////////////////////////////////

#byte PortA= 0x05 // Establece las direcciones de los puertos
#byte PortB= 0x06
#byte PortC= 0x07
#byte PortD= 0x08
#byte PortE= 0x09
```

ARQUIVO FUNCTIONS.H

```
int16 Operacion (int16 Velocidad_Final, int identifiier)
{

switch (identifiier){

case 0: // identificador da suma + 1

if(Velocidad_Final <= 1090)
Velocidad_Final = Velocidad_Final + 10;

break;

case 1: // identificador de suma + 10

if(Velocidad_Final <= 1000)
Velocidad_Final = Velocidad_Final + 100;

break;

case 2: // identificador de resta - 1

if(Velocidad_Final >= 160)
Velocidad_Final = Velocidad_Final - 10;

break;

case 3: // identificador de resta - 10

if(Velocidad_Final >= 250)
Velocidad_Final = Velocidad_Final - 100;

break;

} //end switch

return Velocidad_Final;
} //end Real_Speed_to_Digits

int16 LED_BLIND (int16 count_led, int8 estado)
{
if((estado == 6) || (estado == 4)){
else if((estado == 0) && (count_led <= 50)){
output_low(LED_PROG);
output_low(LED_START);
count_led++;
} else if ((estado == 0) && (count_led > 20) && (count_led < 40)){
output_high(LED_PROG);
output_high(LED_START);
count_led++;
} else if ((estado == 0) && (count_led >= 40)) count_led = 0;
else if ((estado == 1) && (count_led <= 50)) {
output_low(LED_PROG);
```



```

output_high(LED_START);
count_led++;
} else if ((estado == 1) && (count_led > 50) && (count_led <= 100)) {
output_high(LED_PROG);
output_high(LED_START);
count_led++;
} else if ((estado == 1) && (count_led > 100)) count_led = 0;
else if ( (estado == 5) && (count_led < 100)){
output_high(LED_PROG);
output_low(LED_START);
count_led++;
} else if ( (estado == 5) && (count_led >= 100) && (count_led < 200)){
output_high(LED_PROG);
output_high(LED_START);
count_led++;
} else if ( (estado == 5) && (count_led > 200))count_led = 0;
return count_led;

} // end LED_BLINK

////////////////////////////////////
////////////////////////////////////Funcao normal dos segmentos////////////////////////////////////
////////////////////////////////////

void Segmentos (int16 Real_Speed, short int UD) // 100us
{
// A signal CLK is link to RA4 that is an open drain output

int i,j;

for (j=0;j<=3;j++)// ciclo para cada digito
{
output_low(CLK_7S);
delay_cycles(1);
output_low(LOAD);
////////Enderecos dos Registros de Inicializacao //////////
for (i=0; i<=7; i++)
{

if(UD){
if (!bit_test(AD_DIG[j],7-i) output_low(DIN);
else output_high(DIN);
delay_cycles(1);
output_high(CLK_7S);
delay_cycles(1);
output_low(CLK_7S);
} else {
if (!bit_test(AD_DIG[j+4],7-i) output_low(DIN);
else output_high(DIN);
delay_cycles(1);
output_high(CLK_7S);
delay_cycles(1);
output_low(CLK_7S);
}

}

}
}

```

```

///// Dados dos Registros de digitos
for (i=0; i<=7; i++)
{
    if((j==2)&&(i==0)) output_high(DIN);
    else if (i<=3) output_low(DIN);
    else if ((j==0)&&!bit_test(Real_Speed,19-i)) output_low(DIN);
    else if ((j==0)&&bit_test(Real_Speed,19-i)) output_high(DIN);
    else if ((j==1)&&!bit_test(Real_Speed,15-i)) output_low(DIN);
    else if ((j==1)&&bit_test(Real_Speed,15-i)) output_high(DIN);
    else if ((j==2)&&!bit_test(Real_Speed,11-i)) output_low(DIN);
    else if ((j==2)&&bit_test(Real_Speed,11-i)) output_high(DIN);
    else if ((j==3)&&!bit_test(Real_Speed,7-i)) output_low(DIN);
    else if ((j==3)&&bit_test(Real_Speed,7-i)) output_high(DIN);
    delay_cycles(1);
    output_high(CLK_7S);
    delay_cycles(1);
    output_low(CLK_7S);

}
////////// Novo dado a escrever no MAX7219
output_high(LOAD);
delay_cycles(1);

}

}//////////FIM SEGMENTOS

//////////////////////////////////////
//////////////////////////////////////Funcao de inicializacao dos segmentos//////////////////////////////////////
//////////////////////////////////////
boolean In_Segmentos (boolean option, boolean APAGADO)
{

    int i,j;

    if(!option){

        for (j=0;j<=4;j++) // Ciclo para los digitos
        {
            output_low(CLK_7S);
            delay_cycles(1);
            output_low(LOAD);

            //////////Enderecos dos Registros de Inicializacao //////////
            for (i=0; i<=7; i++)
            {
                if (!bit_test(AD_INI[j],7-i)) output_low(DIN);
                else output_high(DIN);
                delay_cycles(1);
                output_high(CLK_7S);
                delay_cycles(1);
                output_low(CLK_7S);
            }
            ////////// Dados dos Registros de Inicializacao

```

```

for (i=0; i<=7; i++)//esta iniciaba en 1
{
    if (!bit_test(DADOS_INI[j],7-i)) output_low(DIN);
    else output_high(DIN);
    delay_cycles(1);
    output_high(CLK_7S);
    delay_cycles(1);
    output_low(CLK_7S);

}

////////// Novo dado a escrever no MAX7219
output_high(LOAD);
delay_cycles(1);

}
Segmentos(ZERO,0); //VEL_MIN
Segmentos(ZERO,1);
APAGADO = FALSE;
} // Inicializacion normal
else {

Segmentos(0xCBDE,0);
Segmentos(0xCBDE,1);

output_low(CLK_7S);
delay_cycles(1);
output_low(LOAD);

//////////Enderecos dos Registros de Inicializacao //////////
for (i=0; i<=7; i++)
{
    if (!bit_test(AD_INI[0],7-i)) output_low(DIN);
    else output_high(DIN);
    delay_cycles(1);
    output_high(CLK_7S);
    delay_cycles(1);
    output_low(CLK_7S);
}
////// Dados dos Registros de Inicializacao
for (i=0; i<=7; i++)//esta iniciaba en 1
{
    if (i<=6) output_low(DIN);
    else if (APAGADO){
        output_high(DIN);
        APAGADO = FALSE;
    }
    else { output_low(DIN);
        APAGADO = TRUE;
    }
    delay_cycles(1);
    output_high(CLK_7S);
    delay_cycles(1);
    output_low(CLK_7S);

}

////////// Novo dado a escrever no MAX7219

```

```

output_high(LOAD);
delay_cycles(1);

}

return APAGADO;
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
}//////////////////////////////////////////////////////////////////END Inicializa_Segmentos

//////////////////////////////////////////////////////////////////Letrura da distância////////////////////////////////////////////////////////////////

int16 distância (boolean module){

int16 distância = 0;

switch (module){

case 0:

output_high(TRIG_I);
delay_us(10);
output_low(TRIG_I);

ECHO1 while(!input(ECHO_I)){if(!input(RST)) goto INICIO;} // leemos el canal de respuesta del sensor

SET_TIMER1(0); // Cuando el sensor responda con UNO activamos el TIMER

while(input(ECHO_I)){if(!input(RST)) goto INICIO;}

distância = GET_TIMER1() - 2; // Velocidad del sonido = 340 m/s
// Prestar atencion al shift right
output_low(TRIG_I);
break;

case 1:

output_high(TRIG_D);
delay_us(10);
output_low(TRIG_D);

ECHO1 while(!input(ECHO_D)){if(!input(RST)) goto INICIO;} // leemos el canal de respuesta del sensor

SET_TIMER1(0); // Cuando el sensor responda con UNO activamos el TIMER

while(input(ECHO_D)){ if(!input(RST)) goto INICIO;}

distância = GET_TIMER1() - 2; // Velocidad del sonido = 340 m/s

output_low(TRIG_D);
break;
} //end switch

INICIO:

```

```

return distância;
} // end distância

////////////////////// FIN de LECTURA DE DISTANCIA ////////////////////////

////////////////////// LETURA DOS PULSES ////////////////////////

//////////////////////

int16 BIN_BCD(int16 binario){

    int i,j,k,X,Y,Z,W;
    int16 BCD = 0;
    boolean bit_BCD;

    X = 0;
    Y = 0;
    Z = 0;
    W = 0;
    // Codigo XCS3 para passar de Binario a BCD 4 digitos

    for( i=0; i<=12; i++){// FOR1

        if (bit_test(binario,13-i) bit_set(BCD,0);
        else bit_clear(BCD,0);

        for (j=0; j<=15; j++){//FOR2

            if ((j<4) && (bit_test(BCD,j))) bit_set(W,j);
            else if ((j<4) && (!bit_test(BCD,j))) bit_clear(W,j);
            else if ((j>=4) && (j<8) && (bit_test(BCD,j))) bit_set(Z,j-4);
            else if ((j>=4) && (j<8) && (!bit_test(BCD,j))) bit_clear(Z,j-4);
            else if ((j>=8) && (j<12) && (bit_test(BCD,j))) bit_set(Y,j-8);
            else if ((j>=8) && (j<12) && (!bit_test(BCD,j))) bit_clear(Y,j-8);
            else if ((j>=12) && (bit_test(BCD,j))) bit_set(X,j-12);
            else if ((j>=12) && (!bit_test(BCD,j))) bit_clear(X,j-12);

        } //END FOR2

        if(X>4) X = X+3;
        if(Y>4) Y = Y+3;
        if(Z>4) Z = Z+3;
        if(W>4) W = W+3;

        for (j=0; j<=15; j++){// FOR3

            if ((j<4) && (bit_test(W,j))) bit_set(BCD,j);
            else if ((j<4) && (!bit_test(W,j))) bit_clear(BCD,j);
            else if ((j>=4) && (j<8) && (bit_test(Z,j-4))) bit_set(BCD,j);
            else if ((j>=4) && (j<8) && (!bit_test(Z,j-4))) bit_clear(BCD,j);
            else if ((j>=8) && (j<12) && (bit_test(Y,j-8))) bit_set(BCD,j);
            else if ((j>=8) && (j<12) && (!bit_test(Y,j-8))) bit_clear(BCD,j);
            else if ((j>=12) && (bit_test(X,j-12))) bit_set(BCD,j);
            else if ((j>=12) && (!bit_test(X,j-12))) bit_clear(BCD,j);

        } //END FOR3
    }
}

```

```

//////////////////////////////////SHIFT LEFT VARIABLES//////////////////////////////////

for (k=0; k <= 14; k++){ //FOR 4

    if(bit_test(BCD,14-k)) bit_BCD = 1;
    else bit_BCD = 0;

    if (bit_BCD) bit_set(BCD,15-k);
    else bit_clear(BCD,15-k);

} // END FOR 4

//////////////////////////////////
} // END_FOR1

if (bit_test(binario,0)) bit_set(BCD,0);
else bit_clear(BCD,0);

return BCD;
}
//////////////////////////////////FIM DO CONVERSOR DE BIN A BCD

//////////////////////////////////INICIO CALCULO DA VELOCIDADE //////////////////////////////////
//////////////////////////////////

void SERVO_DIRECAO(int16 DRA, int16 DLA){

    signed int16 DIR;
    // CONTROL DA DIREÇÃO DO SERVO SAIDAS (IZQ,DER)

    DIR = DLA - DRA;

    if ((DIR >= -16) && (DIR <= 0)) {
        output_low(LED_PROG);
        output_low(LED_START);
    } else if ((DIR > 0) && (DIR <= 16)) {
        output_low(LED_PROG);
        output_low(LED_START);
    } else if (DIR > 16){
        output_high(LED_PROG);
        output_low(LED_START);
    } else if (DIR < -16){
        output_low(LED_PROG);
        output_high(LED_START);
    }
} // END SERVO_DIRECAO ESTADO 4

void SERVO_DIR(int16 DRA, int16 DLA){

    signed int16 DIR;
    // CONTROL DA DIREÇÃO DO SERVO SAIDAS (IZQ,DER)

    DIR = DLA - DRA;

    if ((DIR >= -16) && (DIR <= 0)) {

```

```

    output_high(LS);
    output_high(RS);
} if ((DIR > 0) && (DIR <= 16)) {
    output_high(LS);
    output_high(RS);
} else if (DIR > 16){
    output_high(LS);
    output_low(RS);
} else if (DIR < -16){
    output_low(LS);
    output_high(RS);
}
} // END SERVO_DIRECAO ESTADO 5

```

////////////////////////////////////FIM DA LETURA DOS PULSES////////////////////////////////////

```

/*
(X1,Y1) = (4618,154) -----> (46,18cm/min, 30% Dutty_Cycle)
(X2,Y2) = (7955,219) -----> (79,55cm/min, 42,7% Dutty_Cycle)

```

La pendiente m = 47/2500

La constante b = 65

Para evitar truncamiento de datos se multiplica por 10 la ecuación de ajuste

$$y = (47/2500)*x + 65$$

```

*/
int16 PWM_CT_Calc (int32 Velocidad)
{
    int32 aux = 0;

    aux = Velocidad*233;//Formula experimental 239 teorica 2x + 65
    aux = aux + 60500;
    aux = (aux/1000);

```

```

return aux;
}

```

```

int1 FINDING_FIRST_PULSE(int16 PWM_SET_POINT)
{

```

```

    int1 CTR_P = 0;

```

```

    if (input(PULSE_B)){
        while(input(PULSE_B));
        set_rtcc(0);
        CTR_P = 0;
    } else {
        while(!input(PULSE_B));
        set_rtcc(0);
        CTR_P = 1;
    }

```

```

return CTR_P;
}

```

```

/*
20 pulsos por vuelta, cada vuelta son 18,85cm
Pulso alto 66% pulso bajo 34%

    0,785416 cm  60.000 ms   238005
100 X ----- X ----- = -----
    19,8ms     1 min     counter

*/

int16 Current_measurement(int16 Valor, int8 count_ADC){

if(count_ADC < 128)
Valor = (Valor + read_adc())/2;
else Valor = read_adc();

return Valor;
}

int16 Current_Control (signed int32 ek, int16 mk1){

// 300us ---> Tiempo de muestreo
int32 mk = 0;
signed int32 aux = 0;

aux = ek;

mk = mk1*10;

aux = aux*mk;

aux = aux/100;

aux = aux*9;

aux = aux/1000;

mk = mk + aux;

mk = mk/10;

return mk;
}

void WRITE(int8 data)
{
int8 i;

output_low(SDO);
output_low(CARGA);
delay_us(10);
output_high(SDO);
}

```



```

for(i=0; i<=7; i++){

    delay_us(20);
    if(bit_test(data,i)) output_high(SDO);
    else output_low(SDO);

}
delay_us(20);
output_high(CARGA);
delay_us(20);
output_low(SDO);

}

int16 BCD_BIN(int16 valor){

int16 aux = 0;
int16 uni = 0;
int16 dec = 0;
int16 cent = 0;
int16 mil = 0;

uni = valor & 0x000F;
dec = valor & 0x00F0;
cent = valor & 0x0F00;
mil = valor & 0xF000;

dec = dec >> 4;
cent = cent >> 8;
mil = mil >> 12;

aux = uni + dec*10 + cent*100 + mil*1000;

return aux;
}

int16 SPEED_MEASUREMENT(int1 VEL_INI, int16 vel_medida, int16 Velocidade_Final, int16 c)
{

int32 aux = 0;
int32 counter = 0;

counter = get_rtcc() + c*256;
set_rtcc(0);

aux = 8833242/(counter-376); // 8.833.242,7

if(VEL_INI) vel_medida = (vel_medida + aux)/2;
else vel_medida = aux;

return vel_medida;
}

```

A.2 Código do Microcontrolador PIC18F4550 para o Posicionamento da Tocha de Soldagem

ARQUIVO TORCH.C

```
#include "Torch.h"
#include <float.h>
#include <math.h>

int8 AD_DIG[8] = {AD_D7,AD_D6,AD_D5,AD_D4,AD_D3, AD_D2,AD_D1,AD_D0};
int8 DADOS_INI [5] = {SD, DT, DM, MI, SL};
int8 AD_INI[5] = {AD_SD, AD_DT, AD_DM, AD_MI, AD_SL};
int16 tension_entrada = 24; //Ajustar con el valor de la tension de la fuente de potencia
int8 AUX=0;
int8 estado_M2 = 0;
int8 estado_M1 = 0;
int8 estado_M1_presente = 0;
int8 estado_M2_presente = 0;
int16 PASOS_M2 = 0;
int16 PASOS_M1 = 0;
int16 ROT_SPEED_M2 = SPEED_M2_MAX;
int16 ROT_SPEED_M1 = SPEED_M1_MAX;
int1 direction = 1;
int16 PWMDC1 = 0;
int16 PWMDC2 = 0;
//int16 current_M2A = 0;
//int16 current_M2B = 0;
//int16 current_M1A = 0;
//int16 current_M1B = 0;
int16 VMAX = 0;
int16 VMIN = 512;
int16 FIR00 = 0;
int16 FIR0 = 0;
int16 AUX0 = 0;
int16 Distance = 0;
int16 PASSOS1 = 0;
int16 PASSOS2 = 0;
int16 cos_b = 0;
int16 sen_b = 0;
int16 delta_x;
int16 delta_p;
//int16 Aux1 = 0;
//int16 Aux2 = 0;
//int16 Aux3 = 0;
//int16 Aux4 = 0;
int16 i=0;
int8 order = OFF;
int16 Velocidade_Soldagem = 0;
int1 Read_Speed = 0;
int8 state = 0;
int1 DISABLE_SENSOR = 0;
int16 counter_up = 0;
//int1 OK = FALSE;
//int1 TOGGLE = 0;
```

```

#include "TorchFunctions.h"

#INT_EXT
void ext_isr(void)
{
if(!input(CARGAR))order = READ_BYTE();
}

#INT_TIMER0
void TIMER0_isr(void)
{
Distance = (DIST(FIR00) + Distance)/2;
}

void main()
{
POWER_UP_CONFIG();
tension_entrada = 24; //Ajustar con el valor de la tension de la fuente de potencia
PWMDC1 = PWM_DC1(tension_entrada);
PWMDC2 = PWM_DC2(tension_entrada) + 3;
set_pwm1_duty(PWMDC2/2);
set_pwm2_duty(PWMDC1/2);
sequence_M1(0);
sequence_M2(0);
SET_TIMER0(0);
delay_ms(200);
set_adc_channel(0);
FIR0 = 100000/read_adc();

do{

////////////////////////////////////
////////Maquina de estados de generacion de pasos sin delays

if((PASOS_M1 == 0)){
set_pwm1_duty(PWMDC2/2);
}

if((PASOS_M2 == 0)){
set_pwm2_duty(PWMDC1/2);
}

if(direction){

estado_M1_presente = estado_M1;
estado_M1 = STATE_MACHINE_M1 (estado_M1,PASOS_M1,ROT_SPEED_M1);
PASOS_M1 = PASOS_M1_UPDATE (PASOS_M1,estado_M1_presente,estado_M1);
STEPS_M1 (estado_M1, direction);

if(PASOS_M1 == 0){

estado_M2_presente = estado_M2;
estado_M2 = STATE_MACHINE_M2 (estado_M2,PASOS_M2,ROT_SPEED_M2);

```

```

PASOS_M2 = PASOS_M2_UPDATE (PASOS_M2,estado_M2_presente,estado_M2);
STEPS_M2 (estado_M2, direction);
}

}else if (!direction){

estado_M2_presente = estado_M2;
estado_M2 = STATE_MACHINE_M2 (estado_M2,PASOS_M2,ROT_SPEED_M2);
PASOS_M2 = PASOS_M2_UPDATE (PASOS_M2,estado_M2_presente,estado_M2);
STEPS_M2 (estado_M2, direction);

if(PASOS_M2 == 0){

estado_M1_presente = estado_M1;
estado_M1 = STATE_MACHINE_M1 (estado_M1,PASOS_M1,ROT_SPEED_M1);
PASOS_M1 = PASOS_M1_UPDATE (PASOS_M1,estado_M1_presente,estado_M1);
STEPS_M1 (estado_M1, direction);
}

}

////////////////////FIITRO DO SENSOR DE DISTANCIA IR

if(!DISABLE_SENSOR){
set_adc_channel(0);
AUX0 = FIRO;
FIRO = 50000/read_adc() + FIRO/2;
if((FIRO<(105*AUX0/100))&&(VMAX<FIRO)) VMAX = FIRO;
if((FIRO>(95*AUX0/100))&&(VMIN>FIRO)) VMIN = FIRO;
if(i < 2047)i = i + 1;
else{
i = 0;
FIRO0 = (VMAX + VMIN)/2;
VMAX = FIRO0;
VMIN = FIRO0;
}
}

////////////////////FIN DEL FILTRO DE DISTANCIA

////////////////////CALCULO DE PASOS////////////////////////////////////

if (order == START) {

disable_interrupts(GLOBAL);
DISABLE_SENSOR = 1;

cos_b = cosb(Distance);
sen_b = senb(cos_b);

delta_p = deltap(cos_b);
delta_x = deltax(Distance,cos_b,sen_b);

delay_ms(200);

PASSOS1 = (delta_p)*PASSO_FUSO1;
PASSOS2 = (delta_x)*PASSO_FUSO2;

```

```

PASOS_M1 = PASSOS1;
PASOS_M2 = PASSOS2;

set_pwm1_duty(PWMDC2);
set_pwm2_duty(PWMDC1);

direction = 1;

disable_interrupts(INT_TIMER0);
enable_interrupts(GLOBAL);

order = OFF;

} else if (order == STOP){

delay_ms(500);

direction = 0;

PASOS_M1 = PASSOS1;
PASOS_M2 = PASSOS2;

set_pwm1_duty(PWMDC1/2);
set_pwm2_duty(PWMDC2/2);

enable_interrupts(INT_TIMER0);
DISABLE_SENSOR = 0;

order = OFF;

} else if (order == RESET){
AUX0 = 0;
FIRO = 0;
VMAX = 0;
direction = 1;
PASOS_M2 = 0;
PASOS_M1 = 0;
PASSOS1 = 0;
PASSOS2 = 0;
estado_M1_presente = 0;
estado_M1 = 0;
estado_M2_presente = 0;
estado_M2 = 0;
cos_b = 0;
sen_b = 0;
Velocidade_Soldagem = 0;
Read_Speed = 0;
state = 0;
order = OFF;
DISABLE_SENSOR = 0;
counter_up = 0;
POWER_UP_CONFIG();
tension_entrada = 24; //Ajustar con el valor de la tension de la fuente de potencia
PWMDC1 = PWM_DC1(tension_entrada);
PWMDC2 = PWM_DC2(tension_entrada);
set_pwm1_duty(PWMDC2/2);

```

```

set_pwm2_duty(PWMDC1/2);
    } else if (order == UP_MSG){
        if(PASSOS2 > 200){
            PASSOS2 = PASSOS2 - 200;
            PASOS_M2 = 200;
            direction = 0;
            counter_up = counter_up + 1;
        }
        order = OFF;
    } else if (order == DOWN_MSG){

        if(counter_up > 0){
            PASSOS2 = PASSOS2 + 200;
            PASOS_M2 = 200;
            direction =1;
            counter_up = counter_up - 1;
        }
        order = OFF;
    }
    }while(TRUE);
}
ARQUIVO TORCH.H

```

```

#include <18F4550.h>
#define adc=10

#Fuses HSPLL
#Fuses NOPROTECT,NOBROWNOUT
#Fuses NOWDT,BORV20
#Fuses NOPUT,NOCPD,NOSTVREN
#Fuses NODEBUG,NOLVP,NOWRT,NOWRTD,NOIESO
#Fuses NOFCMEN,NOPBADEN,NOWRTC
#Fuses NOWRTB,NOEBTR,NOEBTRB,NOCPB,NOMCLR
#Fuses NOLPT1OSC
#Fuses NOXINST
#Fuses CPUDIV3
#Fuses NOUSBDIV,NOVREGEN
#Fuses NOICPRT
#Fuses PLL1,CCP2C1

#use delay(clock=32000000)

#define IR_SENSE  PIN_A0 // INPUT 1 PIN 2
#define SENSE_2A  PIN_A1 // INPUT 1 PIN 3
#define SENSE_2B  PIN_A2 // INPUT 1 PIN 4
#define SENSE_1A  PIN_A3 // INPUT 1 PIN 5
#define SENSE_1B  PIN_A4 // INPUT 1 PIN 6
#define TRISA 0xFF

#define CARGAR  PIN_B0 // INPUT 1 PIN 33
#define SDI     PIN_B1 // INPUT 1 PIN 34
#define TEST    PIN_B2 // OUTPUT 0 PIN 35
#define BOTON   PIN_B3 // INPUT 1 PIN 36
#define EN_M1B  PIN_B4 // OUTPUT 0 PIN 37
#define EN_M1A  PIN_B5 // OUTPUT 0 PIN 38

```

```

#define EN_M2B PIN_B6 // OUTPUT 0 PIN 39
#define EN_M2A PIN_B7 // OUTPUT 0 PIN 40
#define TRISB 0x0B

#define CLK_7S PIN_C0 // OUTPUT 0 PIN 15
#define PWM2 PIN_C1 // OUTPUT 0 PIN 16
#define PWM1 PIN_C2 // OUTPUT 0 PIN 17
//PIN_C3 // INPUT 1 PIN 18
//PIN_C4 // INPUT 1 PIN 23
//PIN_C5 // INPUT 1 PIN 24
#define LOAD PIN_C6 // OUTPUT 0 PIN 25
#define DIN PIN_C7 // OUTPUT 0 PIN 26
#define TRISC 0x30

#define IN4_M2 PIN_D0 // OUTPUT 0 PIN 19
#define IN3_M2 PIN_D1 // OUTPUT 0 PIN 20
#define IN2_M2 PIN_D2 // OUTPUT 0 PIN 21
#define IN1_M2 PIN_D3 // OUTPUT 0 PIN 22
#define IN4_M1 PIN_D4 // OUTPUT 0 PIN 27
#define IN3_M1 PIN_D5 // OUTPUT 0 PIN 28
#define IN2_M1 PIN_D6 // OUTPUT 0 PIN 29
#define IN1_M1 PIN_D7 // OUTPUT 0 PIN 30
#define TRISD 0x00

#define FIM_CURSO PIN_E0 // OUTPUT 0 PIN 8
#define INT PIN_E1 // INPUT 1 PIN 9
#define TRISE 0xFF

#define SLAVE_AD 0x68
#define START 0x09
#define STOP 0x05
#define RESET 0xFF
#define OFF 0x00
#define UP_MSG 0x22
#define DOWN_MSG 0x11
#define PASSO_FUSO1 40 // 200 steps por volta(1.8°x half_step) / 5mm por volta
#define PASSO_FUSO2 200 // 200 steps por volta(1.8°x half_step) / 2mm por volta
#define RADIO 10 // radio do bico da tocha
#define CT 240 //comprimento da tocha
#define IRD 71 // Distancia do bico de contato ao carrinho
#define ALPHA 25 //Angulo inicial da tocha
#define COSA 91 // Coseno de alpha x 100

////////// AUXILIAR SIGNALS FOR 7 SEGMENTS

#define SPEED_M1_HIGH 2500 // MOTOR 1 Velocidad 500 pps TIMER 1
#define SPEED_M1_MAX 4000 // MOTOR 1 Velocidad 250 pps TIMER 1
#define SPEED_M1_MIN 25000 // MOTOR 1 Velocidad 40 pps TIMER 1

#define SPEED_M2_HIGH 2500 // MOTOR 2 Velocidad 500 pps TIMER 3
#define SPEED_M2_MAX 4000 // MOTOR 2 Velocidad 250 pps TIMER 3
#define SPEED_M2_MIN 25000 // MOTOR 2 Velocidad 40 pps TIMER 3

// CONSTANTES CONFIGURACAO DO MAX 7219
#define MI 0x0F // Maximum LED intensity 0xA (4 digitos) or 0x0F (8 digitos)
#define SL 0x07 // Scanning Value 0x03 (4 digitos) or 0x07 (8 digitos)

```

```

#define DM 0xFF // Decode Mode 0x0F (4 digitos) 0xFF (8 digitos)
#define SD 0x01 // Normal Mode
#define DT 0x00 // Normal Mode

/////////////////////////////////////////////////////////////////
#define AD_DM 0x09 // Endereco registro de modo do decoder
#define AD_MI 0x0A // Endereco registro de intensidade Maxima do LED
#define AD_SL 0x0B // Endereco registro de limite do escaner
#define AD_SD 0x0C // Endereco registro de modo Shut Down
#define AD_DT 0x0F // Endereco registro do Display Test
/////////////////////////////////////////////////////////////////
#define AD_D0 0x01 // Endereco registro do Digito 0
#define AD_D1 0x02 // Endereco registro do Digito 1
#define AD_D2 0x03 // Endereco registro do Digito 2
#define AD_D3 0x04 // Endereco registro do Digito 3
#define AD_D4 0x05 // Endereco registro do Digito 4
#define AD_D5 0x06 // Endereco registro do Digito 5
#define AD_D6 0x07 // Endereco registro do Digito 6
#define AD_D7 0x08 // Endereco registro do Digito 7
#define CERO 0x0000

#PRIORITY EXT//,TIMER0

```


ARQUIVO TORCHFUNCTIONS.H

```
// Funciona para visualizar los valores en los 7 segmentos
void Segmentos (int16 Real_Speed, short int UD)
{
// A signal CLK is link to RA4 that is an open drain output

int i,j;

for (j=0;j<=3;j++)// ciclo para cada digito
{
output_low(CLK_7S);
delay_cycles(1);
output_low(LOAD);
////////Enderecos dos Registros de Inicializacao //////////
for (i=0; i<=7; i++)
{

if(UD){
if (!bit_test(AD_DIG[j],7-i)) output_low(DIN);
else output_high(DIN);
delay_cycles(1);
output_high(CLK_7S);
delay_cycles(1);
output_low(CLK_7S);
} else {
if (!bit_test(AD_DIG[j+4],7-i)) output_low(DIN);
else output_high(DIN);
delay_cycles(1);
output_high(CLK_7S);
delay_cycles(1);
output_low(CLK_7S);
}
}
}
//////// Dados dos Registros de digitos
for (i=0; i<=7; i++)
{
if((j==2)&&(i==0)) output_high(DIN);
else if (i<=3) output_low(DIN);
else if ((j==0)&&!bit_test(Real_Speed,19-i)) output_low(DIN);
else if ((j==0)&&bit_test(Real_Speed,19-i)) output_high(DIN);
else if ((j==1)&&!bit_test(Real_Speed,15-i)) output_low(DIN);
else if ((j==1)&&bit_test(Real_Speed,15-i)) output_high(DIN);
else if ((j==2)&&!bit_test(Real_Speed,11-i)) output_low(DIN);
else if ((j==2)&&bit_test(Real_Speed,11-i)) output_high(DIN);
else if ((j==3)&&!bit_test(Real_Speed,7-i)) output_low(DIN);
else if ((j==3)&&bit_test(Real_Speed,7-i)) output_high(DIN);
delay_cycles(1);
output_high(CLK_7S);
delay_cycles(1);
output_low(CLK_7S);

}
}
////////// Novo dado a escrever no MAX7219
output_high(LOAD);
delay_cycles(1);
```

```

}

} ////////////////////////////////////////////////////////////////////FIM SEGMENTOS

void In_Segmentos ()
{

    int i,j;

    for (j=0;j<=4;j++) // Ciclo para los digitos
    {
        output_low(CLK_7S);
        delay_cycles(1);
        output_low(LOAD);

        ////////Enderecos dos Registros de Inicializacao ////////
        for (i=0; i<=7; i++)
        {
            if (!bit_test(AD_INI[j],7-i)) output_low(DIN);
            else output_high(DIN);
            delay_cycles(1);
            output_high(CLK_7S);
            delay_cycles(1);
            output_low(CLK_7S);
        }
        ////// Dados dos Registros de Inicializacao
        for (i=0; i<=7; i++)//esta iniciaba en 1
        {
            if (!bit_test(DADOS_INI[j],7-i)) output_low(DIN);
            else output_high(DIN);
            delay_cycles(1);
            output_high(CLK_7S);
            delay_cycles(1);
            output_low(CLK_7S);
        }
        //////// Novo dado a escrever no MAX7219
        output_high(LOAD);
        delay_cycles(1);

    }
    Segmentos(CERO,0);
    Segmentos(CERO,1);

    // Inicializacion normal
    ////////////////////////////////////////////////////////////////////
} ////////////////////////////////////////////////////////////////////END Inicializa_Segmentos

//////////////////////////////////////////////////////////////////

int16 BIN_BCD(int16 binario){

    int i,j,k,X,Y,Z,W;
    int16 BCD = 0;

```

```

boolean bit_BCD;

X = 0;
Y = 0;
Z = 0;
W = 0;
//Codigo XCS3 para passar de Binario a BCD 4 digitos

for( i=0; i<=12; i++){// FOR1

if (bit_test(binario,13-i)) bit_set(BCD,0);
else bit_clear(BCD,0);

for (j=0; j<=15; j++){//FOR2

if ((j<4) && (bit_test(BCD,j))) bit_set(W,j);
else if ((j<4) && (!bit_test(BCD,j))) bit_clear(W,j);
else if ((j>=4) && (j<8) && (bit_test(BCD,j))) bit_set(Z,j-4);
else if ((j>=4) && (j<8) && (!bit_test(BCD,j))) bit_clear(Z,j-4);
else if ((j>=8) && (j<12) && (bit_test(BCD,j))) bit_set(Y,j-8);
else if ((j>=8) && (j<12) && (!bit_test(BCD,j))) bit_clear(Y,j-8);
else if ((j>=12) && (bit_test(BCD,j))) bit_set(X,j-12);
else if ((j>=12) && (!bit_test(BCD,j))) bit_clear(X,j-12);

} //END FOR2

if(X>4) X = X+3;
if(Y>4) Y = Y+3;
if(Z>4) Z = Z+3;
if(W>4) W = W+3;

for (j=0; j<=15; j++){// FOR3

if ((j<4) && (bit_test(W,j))) bit_set(BCD,j);
else if ((j<4) && (!bit_test(W,j))) bit_clear(BCD,j);
else if ((j>=4) && (j<8) && (bit_test(Z,j-4))) bit_set(BCD,j);
else if ((j>=4) && (j<8) && (!bit_test(Z,j-4))) bit_clear(BCD,j);
else if ((j>=8) && (j<12) && (bit_test(Y,j-8))) bit_set(BCD,j);
else if ((j>=8) && (j<12) && (!bit_test(Y,j-8))) bit_clear(BCD,j);
else if ((j>=12) && (bit_test(X,j-12))) bit_set(BCD,j);
else if ((j>=12) && (!bit_test(X,j-12))) bit_clear(BCD,j);

} //END FOR3

//////////////////////SHIFT LEFT VARIABLES//////////////////////////////////////

for (k=0; k <= 14; k++){ //FOR 4

if(bit_test(BCD,14-k)) bit_BCD = 1;
else bit_BCD = 0;

if (bit_BCD) bit_set(BCD,15-k);
else bit_clear(BCD,15-k);

} // END FOR 4

```

```

////////////////////////////////////
} // END_FOR1

if (bit_test(binario,0)) bit_set(BCD,0);
else bit_clear(BCD,0);

return BCD;
}
////////////////////////////////////FIM DO CONVERSOR DE BIN A BCD

int16 PWM_DC1 (int16 TENSION_ENTRADA)
{
int16 DTC = 0;

DTC = (TENSION_ENTRADA*69)/8; //Formula experimental

return DTC;
}

int16 PWM_DC2 (int16 TENSION_ENTRADA)
{
int16 DTC = 0;

DTC = TENSION_ENTRADA*8; //Formula experimental

return DTC;
}

void sequence_M1 (int aux){

switch (aux) {

case 0:// IN1 = 1; IN2 = 0; IN3 = 0; IN4 = 0;

output_high(EN_M1A);
output_high(IN1_M1);
output_low(IN2_M1);

output_low(EN_M1B);
output_low(IN3_M1);
output_low(IN4_M1);

break;

case 1:// IN1 = 0; IN2 = 0; IN3 = 0; IN4 = 1;

output_low(EN_M1A);
output_low(IN1_M1);
output_low(IN2_M1);

output_high(EN_M1B);
output_low(IN3_M1);
output_high(IN4_M1);
}
}

```

```

break;

case 2: // IN1 = 0; IN2 = 0; IN3 = 1; IN4 = 0;

output_low(EN_M1A);
output_low(IN1_M1);
output_low(IN2_M1);

output_high(EN_M1B);
output_high(IN3_M1);
output_low(IN4_M1);

break;

case 4:// IN1 = 0; IN2 = 1; IN3 = 0; IN4 = 0;

output_high(EN_M1A);
output_low(IN1_M1);
output_high(IN2_M1);

output_low(EN_M1B);
output_low(IN3_M1);
output_low(IN4_M1);

break;

case 5:// IN1 = 0; IN2 = 1; IN3 = 0; IN4 = 1;

output_high(EN_M1A);
output_low(IN1_M1);
output_high(IN2_M1);

output_high(EN_M1B);
output_low(IN3_M1);
output_high (IN4_M1);

break;

case 6:// IN1 = 0; IN2 = 1; IN3 = 1; IN4 = 0;

output_high(EN_M1A);
output_low(IN1_M1);
output_high(IN2_M1);

output_high(EN_M1B);
output_high(IN3_M1);
output_low(IN4_M1);

break;

case 8:// IN1 = 1; IN2 = 0; IN3 = 0; IN4 = 0;

output_high(EN_M1A);
output_high(IN1_M1);
output_low(IN2_M1);

```

```

output_low(EN_M1B);
output_low(IN3_M1);
output_low(IN4_M1);

break;

case 9:// IN1 = 1; IN2 = 0; IN3 = 0; IN4 = 1;

output_high(EN_M1A);
output_high(IN1_M1);
output_low(IN2_M1);

output_high(EN_M1B);
output_low(IN3_M1);
output_high(IN4_M1);

break;

case 10:// IN1 = 1; IN2 = 0; IN3 = 1; IN4 = 0;

output_high(EN_M1A);
output_high(IN1_M1);
output_low(IN2_M1);

output_high(EN_M1B);
output_high(IN3_M1);
output_low(IN4_M1);

break;

default : // IN1 = 1; IN2 = 0; IN3 = 1; IN4 = 0;

output_high(EN_M1A);
output_high(IN1_M1);
output_low(IN2_M1);

output_low(EN_M1B);
output_low(IN3_M1);
output_low(IN4_M1);

break;

}
}

void sequence_M2 (int aux){

switch (aux) {

case 0:// IN1 = 1; IN2 = 0; IN3 = 0; IN4 = 0;

output_high(EN_M2A);
output_high(IN1_M2);
output_low(IN2_M2);

output_low(EN_M2B);

```

```

output_low(IN3_M2);
output_low(IN4_M2);

break;

case 1:// IN1 = 0; IN2 = 0; IN3 = 0; IN4 = 1;

output_low(EN_M2A);
output_low(IN1_M2);
output_low(IN2_M2);

output_high(EN_M2B);
output_low(IN3_M2);
output_high(IN4_M2);

break;

case 2: // IN1 = 0; IN2 = 0; IN3 = 1; IN4 = 0;

output_high(EN_M2A);
output_low(IN1_M2);
output_low(IN2_M2);

output_high(EN_M2B);
output_high(IN3_M2);
output_low(IN4_M2);

break;

case 4:// IN1 = 0; IN2 = 1; IN3 = 0; IN4 = 0;

output_high(EN_M2A);
output_low(IN1_M2);
output_high(IN2_M2);

output_low(EN_M2B);
output_low(IN3_M2);
output_low(IN4_M2);

break;

case 5:// IN1 = 0; IN2 = 1; IN3 = 0; IN4 = 1;

output_high(EN_M2A);
output_low(IN1_M2);
output_high(IN2_M2);

output_high(EN_M2B);
output_low(IN3_M2);
output_high (IN4_M2);

break;

case 6:// IN1 = 0; IN2 = 1; IN3 = 1; IN4 = 0;

output_high(EN_M2A);

```

```

output_low(IN1_M2);
output_high(IN2_M2);

output_high(EN_M2B);
output_high(IN3_M2);
output_low(IN4_M2);

break;

case 8:// IN1 = 1; IN2 = 0; IN3 = 0; IN4 = 0;

output_high(EN_M2A);
output_high(IN1_M2);
output_low(IN2_M2);

output_low(EN_M2B);
output_low(IN3_M2);
output_low(IN4_M2);

break;

case 9:// IN1 = 1; IN2 = 0; IN3 = 0; IN4 = 1;

output_high(EN_M2A);
output_high(IN1_M2);
output_low(IN2_M2);

output_high(EN_M2B);
output_low(IN3_M2);
output_high(IN4_M2);

break;

case 10:// IN1 = 1; IN2 = 0; IN3 = 1; IN4 = 0;

output_high(EN_M2A);
output_high(IN1_M2);
output_low(IN2_M2);

output_high(EN_M2B);
output_high(IN3_M2);
output_low(IN4_M2);

break;

default : // IN1 = 1; IN2 = 0; IN3 = 1; IN4 = 0;

output_high(EN_M2A);
output_high(IN1_M2);
output_low(IN2_M2);

output_low(EN_M2B);
output_low(IN3_M2);
output_low(IN4_M2);

break;

```



```

}
}

void STEPS_M1 (int8 estado_M1, int1 direction)
{

switch(estado_M1){

case 0:

sequence_M1(0);

break;

// AB -> B -> AN_B -> AN -> AN_BN -> BN -> A_BN -> A
case 1:

if(direction) sequence_M1(10); // AB
else if(!direction) sequence_M1(8); // A
break;

case 2:

if(direction) sequence_M1(2); // B
else if(!direction) sequence_M1(9); // A_BN

break;

case 3:

if(direction) sequence_M1(6); // AN_B
else if(!direction) sequence_M1(1); // BN
break;

case 4:

if(direction) sequence_M1(4); // AN
else if(!direction) sequence_M1(5); // AN_BN

break;

case 5:

if(direction) sequence_M1(5); // AN_BN
else if(!direction) sequence_M1(4); // AN

break;

case 6:

if(direction) sequence_M1(1); // BN
else if(!direction) sequence_M1(6); // AN_B

break;

```

```

case 7:

if(direction) sequence_M1(9); // A_BN
else if(!direction) sequence_M1(2); // B

break;

case 8:

if(direction) sequence_M1(8); // A_BN
else if(!direction) sequence_M1(10); // AB

break;

default:
sequence_M1(0);
break;

}
}

int16 PASOS_M1_UPDATE (int16 PASOS_M1, int8 estado_M1_presente, int8 estado_M1){

if((estado_M1_presente != estado_M1) && (estado_M1 != 0)) {
PASOS_M1 = PASOS_M1 - 1;
}
return PASOS_M1;
}

int8 STATE_MACHINE_M1 (int8 estado_M1,int16 PASOS_M1, int16 ROT_SPEED_M1){

if(PASOS_M1 == 0){
estado_M1 = 0;
}
else if((PASOS_M1 > 0) && (estado_M1 == 0)) {
set_timer1(0);
estado_M1 = 1;

} else if((get_timer1() > ROT_SPEED_M1) && (estado_M1 < 8)){

set_timer1(0);
estado_M1 = estado_M1 + 1;

} else if ((get_timer1() > ROT_SPEED_M1) && (estado_M1 == 8)){
set_timer1(0);
estado_M1 = 1;
}

return estado_M1;
}

void STEPS_M2 (int estado_M2, int1 direction)
{

switch(estado_M2){

```

```

case 0:

sequence_M2(0);

break;

// AB -> B -> AN_B -> AN -> AN_BN -> BN -> A_BN -> A
case 1:

if(direction) sequence_M2(10); // AB
else if(!direction) sequence_M2(8); // A
break;

case 2:

if(direction) sequence_M2(2); // B
else if(!direction) sequence_M2(9); // A_BN

break;

case 3:

if(direction) sequence_M2(6); // AN_B
else if(!direction) sequence_M2(1); // BN
break;

case 4:

if(direction) sequence_M2(4); // AN
else if(!direction) sequence_M2(5); // AN_BN

break;

case 5:

if(direction) sequence_M2(5); // AN_BN
else if(!direction) sequence_M2(4); // AN

break;

case 6:

if(direction) sequence_M2(1); // BN
else if(!direction) sequence_M2(6); // AN_B

break;

case 7:

if(direction) sequence_M2(9); // A_BN
else if(!direction) sequence_M2(2); // B

break;

case 8:

```

```

if(direction) sequence_M2(8); // A_BN
else if(!direction) sequence_M2(10); // AB

break;

default:
sequence_M2(0);
break;

}

}

int8 STATE_MACHINE_M2 (int8 estado_M2, int16 PASOS_M2,int16 ROT_SPEED_M2){

if(PASOS_M2 == 0){
estado_M2 = 0;
}
else if((PASOS_M2 > 0) && (estado_M2 == 0)) {

set_timer3(0);
estado_M2 = 1;

} else if((get_timer3() > ROT_SPEED_M2) && (estado_M2 < 8)) {

set_timer3(0);
estado_M2 = estado_M2 + 1;

} else if ((get_timer3() > ROT_SPEED_M2) && (estado_M2 == 8)) {
set_timer3(0);
estado_M2 = 1;
}

return estado_M2;
}

int16 PASOS_M2_UPDATE (int16 PASOS_M2, int8 estado_M2_presente, int8 estado_M2){

if((estado_M2_presente != estado_M2) && (estado_M2 != 0)){
PASOS_M2 = PASOS_M2 - 1;
//Segmentos(BIN_BCD(PASOS_M2),1);
}
return PASOS_M2;
}

void POWER_UP_CONFIG(){

setup_adc_ports(AN0_TO_AN4|VSS_VDD); //
setup_adc(ADC_CLOCK_DIV_64); // Tad = 2us
setup_psp(PSP_DISABLED);
setup_wdt(WDT_OFF);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_128 );// Freq -> 62.500 kHz (16us) 16 bits OFLOW = 1
segundo
setup_timer_1(T1_INTERNAL|T1_DIV_BY_8); // Freq -> 1MHz (1us) 5000 conteos minimos para
5ms o 200 pulsos por segundo

```

```

        setup_timer_3(T3_INTERNAL|T3_DIV_BY_8); // Freq -> 1MHz (1us) 5000 conteos minimos para
5ms o 200 pulsos por segundo
        setup_timer_2(T2_DIV_BY_4,39,1); // Freq PWM -> 50kHz (Period = 20us)
        setup_ccp2(CCP_PWM);
        setup_ccp1(CCP_PWM);

        setup_comparator(NC_NC_NC_NC);
        setup_vref(FALSE);
        setup_spi(SPI_SS_DISABLED);

        enable_interrupts(global);
        enable_interrupts(INT_EXT);
        EXT_INT_EDGE(H_TO_L);
        enable_interrupts(INT_TIMER0);

        setup_oscillator(OSC_32MHZ|OSC_PLL_ON);

        Set_tris_A(TRISA); // TRISA
        Set_tris_B(TRISB); // TRISB
        Set_tris_C(TRISC); // TRISC
        Set_tris_D(TRISD); // TRISD
        Set_tris_E(TRISE); // TRISE
        // Inicializacion de todas las salidas y sub-sistemas

        delay_ms(100);

        //output_high(TEST);

        output_low(EN_M2B);
        output_low(IN4_M2);
        output_low(IN3_M2);

        output_low(EN_M2A);
        output_low(IN2_M2);
        output_low(IN1_M2);

        output_low(EN_M1B);
        output_low(IN4_M1);
        output_low(IN3_M1);

        output_low(EN_M1A);
        output_low(IN2_M1);
        output_low(IN1_M1);

        delay_ms(100);

        //In_Segmentos();

    }

    int16 DIST(int16 FIR00){

        int16 Valor = 0;

        if(FIR00 > 472) Valor = 0;
        else if((FIR00 >= 463) && (FIR00 < 472)) Valor = 240;

```

```

else if((FIR00 >= 453) && (FIR00 < 463)) Valor = 235;
else if((FIR00 >= 443) && (FIR00 < 453)) Valor = 230;
else if((FIR00 >= 430) && (FIR00 < 443)) Valor = 225;
else if((FIR00 >= 417) && (FIR00 < 430)) Valor = 220;
else if((FIR00 >= 409) && (FIR00 < 417)) Valor = 215;
else if((FIR00 >= 402) && (FIR00 < 409)) Valor = 210;
else if((FIR00 >= 396) && (FIR00 < 402)) Valor = 205;
else if((FIR00 >= 391) && (FIR00 < 396)) Valor = 200;
else if((FIR00 >= 384) && (FIR00 < 391)) Valor = 195;
else if((FIR00 >= 378) && (FIR00 < 384)) Valor = 190;
else if((FIR00 >= 369) && (FIR00 < 378)) Valor = 185;
else if((FIR00 >= 360) && (FIR00 < 369)) Valor = 180;
else if((FIR00 >= 350) && (FIR00 < 360)) Valor = 175;
else if((FIR00 >= 340) && (FIR00 < 350)) Valor = 170;
else if((FIR00 >= 331) && (FIR00 < 340)) Valor = 165;
else if((FIR00 >= 322) && (FIR00 < 331)) Valor = 160;
else if((FIR00 >= 310) && (FIR00 < 322)) Valor = 155;
else if((FIR00 >= 298) && (FIR00 < 310)) Valor = 150;
else if((FIR00 >= 289) && (FIR00 < 298)) Valor = 145;
else if((FIR00 >= 289) && (FIR00 < 298)) Valor = 140;
else if((FIR00 >= 268) && (FIR00 < 289)) Valor = 135;
else if((FIR00 >= 255) && (FIR00 < 268)) Valor = 130;
else if((FIR00 >= 245) && (FIR00 < 255)) Valor = 125;
else if((FIR00 >= 235) && (FIR00 < 245)) Valor = 120;
else if((FIR00 >= 224) && (FIR00 < 235)) Valor = 115;
else if((FIR00 >= 214) && (FIR00 < 224)) Valor = 110;
else if((FIR00 >= 205) && (FIR00 < 214)) Valor = 105;
else if((FIR00 >= 196) && (FIR00 < 205)) Valor = 100;
else if((FIR00 >= 188) && (FIR00 < 196)) Valor = 95;
else if((FIR00 >= 180) && (FIR00 < 188)) Valor = 90;
else if((FIR00 >= 170) && (FIR00 < 180)) Valor = 85;
else if((FIR00 >= 159) && (FIR00 < 170)) Valor = 80;
else if((FIR00 >= 153) && (FIR00 < 159)) Valor = 75;
else if((FIR00 >= 148) && (FIR00 < 152)) Valor = 70;

return Valor;
}

```

```
int16 cosb (int16 Distance){
```

```

int32 Det;
int32 Dm2 = 0;
int32 b = 0;
int32 c = 0;
int32 aux = 0;
int16 cos_b;

```

```

Dm2 = Distance + 25;
Dm2 = Dm2*Dm2;
aux = Dm2 + 11450;

```

```

b = 973700/aux;
c = 100*(Dm2 - 8280);
c = c/aux;

```

```
Det = b*b;
```

```

Det = Det + 100*c;
Det = (sqrt(Det) - b);
cos_b = Det;

return cos_b;
}

int16 senb (int16 cos_b){

int16 sen_b;
int32 temp = 0;
int32 temp2 = 0;

temp = cos_b;
temp = temp*temp;
temp2 = 10000 - temp;
temp2 = sqrt(temp2);
sen_b = temp2;

return sen_b;
}

int16 deltap (int16 cos_b){

int16 aux = 0;
float temp = 0;
float temp2 = 0;
float temp3 = 0;

temp = cos_b;
temp = temp/100;

temp3 = 118 - 57*temp;

temp2 = 118 - 57*temp - sqrt(temp3*temp3 + 13450*temp - 12550);

aux = floor(temp2);

return aux;
}

int16 deltax (int16 Distance, int16 cos_b, int16 sen_b) {

int16 delta_x = 0;
int32 temp = 0;
int32 temp2 = 0;
int32 temp3 = 0;
int32 temp4 = 0;
int32 temp5 = 0;

// (100*(D_m - 110*cos_b - 41*sin_b - 11*cos_b*cos_b/sen_b))/(94*cos_b + 36*sin_b)
temp4 = (cos_b*cos_b)/sen_b;
temp4 = 110*temp4;
temp3 = sen_b;
temp3 = 410*temp3;
temp2 = cos_b;

```

```

temp2 = 1080*temp2;
temp5 = 94*cos_b+36*sen_b;
temp = Distance + 5; ////////////////////////////////////////////////////
**%*%*%*%*%$*%*#*$*#*&*%&*%&$*%&$*%&%%*$*%&*%&$*%&*%&$*%&*%&$*%&*%&$*%&*
temp = 1000*temp;
temp = temp - temp2 - temp3 - temp4;
temp = temp*10;
temp = temp/temp5;

delta_x = temp - 5; ////////////////////////////////////////////////////
**%*%*%*%*%$*%*#*$*#*&*%&*%&$*%&$*%&%%*$*%&*%&$*%&*%&$*%&*%&$*%&*%&$*%&*%&$*%&*
return delta_x;

}

int8 READ_BYTE(){

int8 i,dato=0;

while(!input(SDI));
delay_us(10);

for(i=0; i<=7; i++)
{
delay_us(20);
if(input(SDI)) bit_set(dato,i);
else bit_clear(dato,i);
}
return dato;
}

```


A.3 Código do Microcontrolador PIC10F322 para o Controle da Direção

```
ARQUIVO SERVO_MAIN.C

/*
 * File: Servo_main.c
 * Author: Leonardo
 *
 * Created on 25 de febrero de 2014, 01:31 PM
 */

#include <htc.h>
#include "config.h"
// #include "delay.c"
// #include "delay.h"
#include "io.h"
#define _XTAL_FREQ 2MHZ
#define CT_MIN 48 // 49
#define CT_MID 54 // 55
#define CT_MAX 59 // 62
#define CT_100 622

__CONFIG(CONFIG_REG);

void SET_PWM(unsigned short value);

void main(void) {

    INTCON = 0x88; //1000 1000 Interrupt Register Configuration
    PIE1 = 0x00; //0000 0010 Peripheral Interrupt Enable Register 1
    PIR1 = 0x00; //0000 0000 Peripheral Interrupt Request Register 1
    OSCCON = 0x40; // FOSC = 2MHz OSCCON = 0100 0000
    CLKRCON = 0x00;
    ANSELA = 0x00; // 0000 0000 Port Analog Selection Register
    TRISA = 0x0D; // 0000 1101 Only RA1 is an output
    IOCAP = 0x09; // 0000 1001 ENABLE Positive Edge
    IOCAN = 0x09; // 0000 1001 ENABLE Negative Edge
    IOCAF = 0; // Falling/Rising Edge detector on A or B port
    //
    // FREQUENCY OF PWM = (PR2+1)*(4*(1/2MHz))*(TMR2_PRE-SCALE) = 50HZ
    // Resolution = log(4*(PR2+1))/log(2)
    PWM2CON = 0x00; ///PWM2EN = 1; //PWM2OE = 1; //PWM2POL = 0;
    PR2 = 155; // PR2 ---> 155 decimal
    T2CON = 0x07; // 0000 0111 PRE-SCALE -> 1:64 T2CKPS<1:0> == 11
    PWM2DCL = 0;
    PWM2DCH = 0;
    PWM2CON = 0xC0; /// 1100 000 PWM2EN = 1; //PWM2OE = 1; //PWM2POL = 0;

    SET_PWM(CT_MID);

    while(1){

        if(IOCIF){// Interrup-on-change Interrupt Flag bit
            IOCAF = 0;
```

```

    if(!RS && LS) SET_PWM(CT_MIN);
    else if(RS && !LS) SET_PWM(CT_MAX);
    else SET_PWM(CT_MID);
  }

}
}

```

```

void SET_PWM(unsigned short value){

```

```

    PWM2DCH = value >> 2;
    PWM2DCL = value << 6;

```

```

} // FIM SET_PWM

```

ARQUIVO IO.H

```

/*
 * File: io.h
 * Author: Leonardo
 *
 * Created on 25 de febrero de 2014, 11:13 PM
 */

```

```

#ifndef IO_H
#define IO_H

```

```

#ifndef BITNUM
#define BITNUM(adr, bit) ((unsigned)&adr)*8+(bit) //-- used for port defs
#endif

```

```

static bit PWM2 @ BITNUM(PORTA,1); //PIN 4
static bit T_PWM2 @ BITNUM(TRISA,1);

```

```

static bit LS @ BITNUM(PORTA,0); //PIN 5
static bit T_LS @ BITNUM(TRISA,0);

```

```

static bit RS @ BITNUM(PORTA,3); //PIN 8
static bit T_RS @ BITNUM(TRISA,3);

```

```

#endif /* IO_H */

```

ARQUIVO CONFIG.H

```

*
 * Created on 25 de febrero de 2014, 01:34 PM
 */

```

```

#ifndef CONFIG_H
#define CONFIG_H

```

```

#define CONFIG_REG
FOSC_INTOSC&WDTE_OFF&PWRTE_OFF&MCLRE_OFF&CP_OFF&BOREN_ON&WRT_OFF&LPBOR_OFF&LVP_OF
F

```

```

/*
#define bit_set(var,bitno) ((var) |= 1<<(bitno))

```

```
#define bit_clr(var,bitno) ((var) &= ~(1<<(bitno)))
#define bit_test(var,bitno) ((var) & (1<<(bitno)))
*/
#endif      /* CONFIG_H */
```

APÊNDICE B

B.1 Código de MatLab™ Programa para o Projeto Mecânico

ARQUIVO WRLRA.M

```
function varargout = WRLRA(varargin)
% WRLRA MATLAB code for WRLRA.fig
%   WRLRA, by itself, creates a new WRLRA or raises the existing
%   singleton*.
%
%   H = WRLRA returns the handle to a new WRLRA or the handle to
%   the existing singleton*.
%
%   WRLRA('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in WRLRA.M with the given input arguments.
%
%   WRLRA('Property','Value',...) creates a new WRLRA or raises
%   the existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before WRLRA_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to WRLRA_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help WRLRA

% Last Modified by GUIDE v2.5 02-Dec-2014 00:44:03

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @WRLRA_OpeningFcn, ...
                  'gui_OutputFcn', @WRLRA_OutputFcn, ...
                  'gui_LayoutFcn', [], ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before WRLRA is made visible.
function WRLRA_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to WRLRA (see VARARGIN)

% Choose default command line output for WRLRA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

initialize_gui(hObject, handles, false);

% UIWAIT makes WRLRA wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = WRLRA_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function CT_CreateFcn(hObject, eventdata, handles)
% hObject handle to CT (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function CT_Callback(hObject, eventdata, handles)
% hObject handle to CT (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of CT as text
% str2double(get(hObject,'String')) returns contents of CT as a double
CT = str2double(get(hObject, 'String'));
if isnan(CT)
    set(hObject, 'String', 0);
    errorDlg('Input must be a number', 'Error');
end

handles.data.CT = CT;
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.

```

```

function Diametro_CreateFcn(hObject, eventdata, handles)
% hObject handle to Diametro (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function Diametro_Callback(hObject, eventdata, handles)
% hObject handle to Diametro (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Diametro as text
% str2double(get(hObject,'String')) returns contents of Diametro as a double

```

```

Diametro = str2double(get(hObject, 'String'));
if isnan(Diametro)
    set(hObject, 'String', 0);
    errorDlg('Input must be a number', 'Error');
end

```

```

handles.data.Diametro = Diametro;
guidata(hObject,handles)

```

```

% --- Executes during object creation, after setting all properties.
function Radio_CreateFcn(hObject, eventdata, handles)
% hObject handle to Radio (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function Radio_Callback(hObject, eventdata, handles)
% hObject handle to Radio (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Radio as text
% str2double(get(hObject,'String')) returns contents of Radio as a double

```

```

Radio = str2double(get(hObject, 'String'));
if isnan(Radio)
    set(hObject, 'String', 0);
    errorDlg('Input must be a number', 'Error');
end

```

```

handles.data.Radio = Radio;
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function Altura_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Altura (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Altura_Callback(hObject, eventdata, handles)
% hObject    handle to Altura (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Altura as text
%       str2double(get(hObject,'String')) returns contents of Altura as a double

Altura = str2double(get(hObject,'String'));
if isnan(Altura)
    set(hObject, 'String', 0);
    errorlg('Input must be a number', 'Error');
end

handles.data.Altura = Altura;
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function alfa_CreateFcn(hObject, eventdata, handles)
% hObject    handle to alfa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function alfa_Callback(hObject, eventdata, handles)
% hObject    handle to alfa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of alfa as text
%       str2double(get(hObject,'String')) returns contents of alfa as a double

alfa = str2double(get(hObject, 'String'));

```



```

if isnan(alfa)
    set(hObject, 'String', 0);
    errorDlg('Input must be a number', 'Error');
end

handles.data.alfa = alfa;
guidata(hObject,handles)

% --- Executes on button press in calculate.
function calculate_Callback(hObject, eventdata, handles)
% hObject   handle to calculate (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

R = (handles.data.Diametro) / 2;
A = 70 + R;
alpha = handles.data.alfa*pi/180;
dsm = 62.13*cos(alpha);
x = handles.data.CT*cos(alpha);
IRD = x - dsm;
Dm_min = IRD + handles.data.Radio*sin(alpha);
Dm = Dm_min:5:215;
a = 68;
b = 57;
c = b*cos(alpha) + ((a^2) - (b^2)*(sin(alpha))^2)^(0.5);

cosb = cos_b(Dm,handles.data.Altura,A);
senb = sen_b(cosb);
dx_prima = dx_p(cosb,a,b,c);
delta = del(Dm,alpha,cosb,senb,handles.data.CT,handles.data.Radio);

Beta = (acos(cosb).*180)./pi;

plot(handles.seno_axes,Dm,cosb,Dm,senb,'-
r.','LineWidth',3,'MarkerEdgeColor','y','MarkerFaceColor','k','MarkerSize',5);
grid(handles.seno_axes,'on');
set(handles.seno_axes,'XLim',[min(Dm) max(Dm)]);
set(handles.seno_axes,'YLim',[min(cosb) 1]);
title(handles.seno_axes,'Funções seno e coseno'),
xlabel(handles.seno_axes,'Distância Dm [mm]')
ylabel(handles.seno_axes,'sen(verm.) e cos(azul)');

plot(handles.angulo_axes,Dm,Beta,'-
k.','LineWidth',3,'MarkerEdgeColor','c','MarkerFaceColor','k','MarkerSize',5);
grid(handles.angulo_axes,'on');
set(handles.angulo_axes,'XLim',[min(Dm) max(Dm)]);
set(handles.angulo_axes,'YLim',[min(Beta) max(Beta)]);
title(handles.angulo_axes,'Função ângulo \beta')
xlabel(handles.angulo_axes,'Distância Dm [mm]')
ylabel(handles.angulo_axes,'\delta [mm]')

plot(handles.Delta_axes,Dm,delta,'-
k.','LineWidth',3,'MarkerEdgeColor','r','MarkerFaceColor','k','MarkerSize',5);
grid(handles.Delta_axes,'on');
set(handles.Delta_axes,'XLim',[min(Dm) max(Dm)]);
set(handles.Delta_axes,'YLim',[min(delta) max(delta)]);

```

```

title(handles.Delta_axes,'Função \delta')
xlabel(handles.Delta_axes,'                               Distância Dm [mm]')
ylabel(handles.Delta_axes,'\delta [mm]')

plot(handles.delta_prima_axes,Dm,dx_prima,'-
k.','LineWidth',3,'MarkerEdgeColor','g','MarkerFaceColor','k','MarkerSize',5);
grid(handles.delta_prima_axes,'on');
set(handles.delta_prima_axes,'XLim',[min(Dm) max(Dm)]);
set(handles.delta_prima_axes,'YLim',[min(dx_prima) max(dx_prima)]);
title(handles.delta_prima_axes,'Função \Delta\prime')
xlabel(handles.delta_prima_axes,'                               Distância Dm [mm]')
ylabel(handles.delta_prima_axes,'\Delta\prime [mm]')

% --- Executes on button press in reset.
function reset_Callback(hObject, eventdata, handles)
% hObject    handle to reset (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

initialize_gui(gcf, handles, true);

% -----
function initialize_gui(fig_handle, handles, isreset)
% If the metricdata field is present and the reset flag is false, it means
% we are we are just re-initializing a GUI by calling it from the cmd line
% while it is up. So, bail out as we dont want to reset the data.

handles.data.CT = 180;
handles.data.Diametro = 37;
handles.data.Altura = 100;
handles.data.Radio = 11;
handles.data.alfa = 21;

set(handles.CT, 'String', handles.data.CT);
set(handles.Diametro, 'String', handles.data.Diametro);
set(handles.Altura, 'String', handles.data.Altura);
set(handles.Radio, 'String', handles.data.Radio);
set(handles.alfa, 'String', handles.data.alfa);

% Update handles structure
guidata(handles.figure1, handles);

% --- Executes on key press with focus on CT and none of its controls.
function CT_KeyPressFcn(hObject, eventdata, handles)

```

ARQUIVO COS_B.M

```
function cosb = cos_b(Dm,H,A)
```

```
b = ((2*A*H)/(H^2 +(Dm+25).^2));  
c = ((Dm+25).^2 - A^2)/(H^2 + (Dm+25).^2 );
```

```
Determinante = b.^2 + 4.*c;
```

```
cosb = (sqrt(Determinante) - b)./2;
```

```
return
```

ARQUIVO DEL.M

```
function delta = del(Dm,alpha,cosb,senb,CT,r)
```

```
aux1 = cos(alpha)*(CT-62.13);  
aux2 = sin(alpha)*(CT-62.13);  
aux3 = r.*(cosb.^2);  
aux4 = cosb.*cos(alpha);  
aux5 = senb.*sin(alpha);  
aux6 = aux4 + aux5;
```

```
delta = (Dm - aux1.*cosb - aux2.*senb - aux3./senb + 5) ./ (aux4 + aux5);
```

```
return
```

ARQUIVO DX_P.M

```
function dx_prima = dx_p(cosb,a,b,c)
```

```
dx_prima = c - b.*cosb - (4*(c - b.*cosb).^2 + 4.*(2.*b.*c.*cosb - c^2 + a^2 - b^2)).^(0.5)./2 + 5;
```

```
return
```

ARQUIVO SEN_B.M

```
function senb = sen_b(cosb)
```

```
senb = (1-cosb.^2).^(0.5);
```

```
return
```

B.2 Código de MatLab™ para a Transformada de Fourier

PROCESAMENTO DAS SINAIS DE TENSÃO E CORRENTE MEDIDAS

PROGRAMA PARA PROCESSAR OS DADOS DOS ARQUIVOS TXT

```
clc;
clear all;
%App diretorio
%/Users/leonardorivasarce/Documents/MATLAB/Measurements
extention = '*.txt';
%[file_name,file_path] = uigetfile('*.*','OPEN FILE'); %Alternativa
%grafico
file_path='/Users/leonardorivasarce/Documents/Lapes/'; %Pasta onde est,,o
%os arquivos a utilizar
file_name='Teste000.txt';% Formato do nome

path = strcat(file_path,extention);% Concatenate path with extention txt
file = dir(path); %Procura o numero de arquivos no diretorio
numfiles = length(file); % Determina o tamanho das variables o arquivos

for i=1:numfiles

file_name = file(i).name; %Re-nomea cada ciclo
measurements = [file_path,file_name]; % Cria o nome do arquivo a carregar
data = load(measurements); % Carrega o arquivo no ordem sequencial

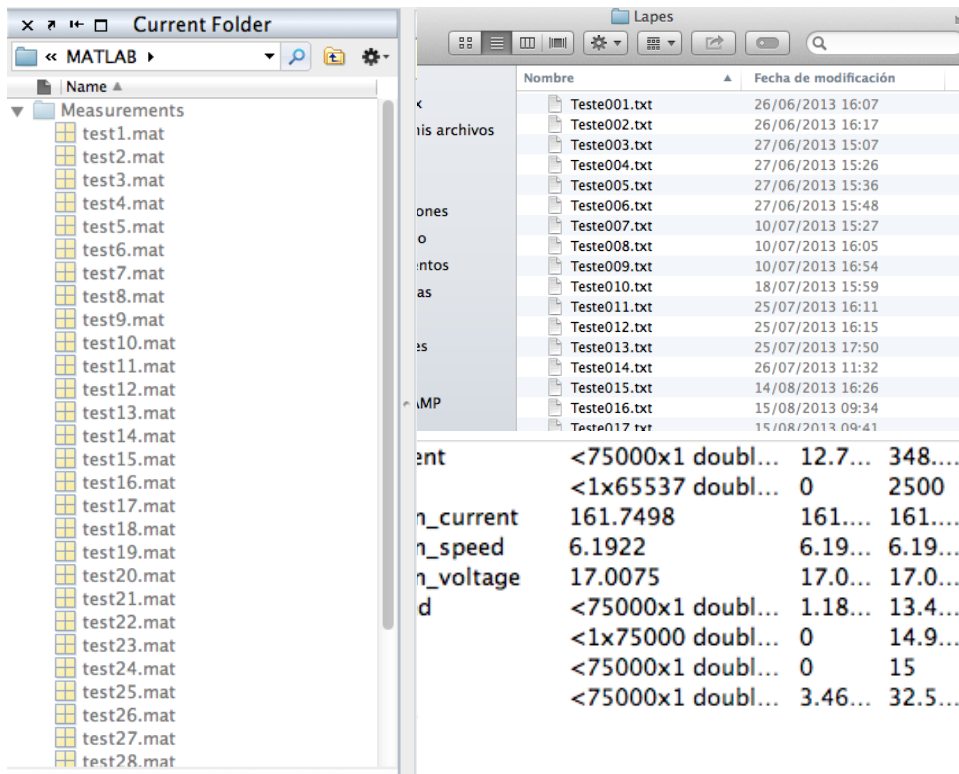
weld_matrix = data(:,1:4);

time = weld_matrix(:,1); %Variavel tempo nas mediÁies
volt = weld_matrix(:,2); %Variavel tempo nas mediÁies
current = weld_matrix(:,3); %Variavel tempo nas mediÁies
speed = weld_matrix(:,4); %Variavel tempo nas mediÁies
means = mean(weld_matrix);

mean_voltage = means(1,2);
mean_current = means(1,3);
mean_speed = means(1,4);

myfile = sprintf('/Users/leonardorivasarce/Documents/MATLAB/Measurements/test%d',i);
save(myfile, 'time', 'volt', 'current', 'speed', 'mean_voltage', 'mean_current', 'mean_speed');

end
```



Todos os arquivos das medições foram armazenados numa pasta chamada Lapes e ao aplicar o programa feito acima, este importou todos os dados ao Workspace de Matlab com os nomes *test_xx* (cada arquivo contendo por separado as variáveis de corrente, tensão, velocidade e tempo).

Todo isto para aplicar os controles e filtros de análise das variáveis, como por exemplo a transformada FFT para enxergar as frequências espectrais das sinais e a sua densidades de potência.

Qualquer arquivo de medição futuro somente terá que se copiar na pasta Lapes e automaticamente serão aplicados os controles de análise.

TRANSFORMADA DE FOURIER PARA ENXERGAR O ESPECTRO DE POTÊNCIA DA SINAIL

```
Fs = 5000;           % Sampling frequency
T = 1/Fs;           % Sample time
```

```
L = length(volt);   % Length of signal
t = (0:L-1)*T;      % Time vector
plot(Fs*t(1:50),volt(1:50))
title('Signal Corrupted with Zero-Mean Random Noise')
xlabel('time (milliseconds)')
```

```
NFFT = 2^nextpow2(L); % Next power of 2 from length of y
Y = fft(volt,NFFT)/L;
f = Fs/2*linspace(0,1,NFFT/2+1);
```

```
% Plot single-sided amplitude spectrum.
plot(f,2*abs(Y(1:NFFT/2+1)))
title('Single-Sided Amplitude Spectrum of y(t)')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')
```

