



UNIVERSIDADE FEDERAL DE MINAS GERAIS

**PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA MECÂNICA**

**REMANUFATURA DE MANIPULADORES ROBÓTICOS
INDUSTRIAIS UTILIZANDO ARQUITETURA ABERTA**

MARCELO HENRIQUE SOUZA BOMFIM

Belo Horizonte, 21 de janeiro de 2013.

MARCELO HENRIQUE SOUZA BOMFIM

**REMANUFATURA DE MANIPULADORES ROBÓTICOS
INDUSTRIAIS UTILIZANDO ARQUITETURA ABERTA**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Mestre em Engenharia Mecânica.

Área de concentração: Processos de Fabricação

Orientador: Prof. Dr. Eduardo José Lima II

Departamento de Engenharia Mecânica, UFMG

BELO HORIZONTE - MG

2013

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter iluminado meu caminho nos momentos difíceis, ter me dado forças e perseverança para atingir meu objetivo de fazer o manipulador robótico ASEA operar.

Aos meus pais, por mostrarem que é através do trabalho que conseguimos alcançar nossos objetivos. Aos meus irmãos, pela paciência e incentivo. Aos meus amigos e primos pela amizade, momentos de diversão e apoio.

À Clarinha, pelo amor, incentivo, paciência e dedicação. Por estar sempre ao meu lado, ajudando nas correções ortográficas e ofertando sugestões que tanto engrandeceram este trabalho.

Ao eterno mestre Eduardo José Lima II que, mais que um orientador, é um grande amigo, e apesar de sua mudança e ainda estar com uma filha recém nascida, mostrou-se sempre à disposição, incentivando, apoiando e auxiliando nos testes experimentais da trajetória do robô.

Ao Professor Alexandre Queiroz Bracarense, por ter acreditado em meu trabalho e por ter oferecido toda a estrutura necessária para que colocasse minha dissertação em prática, sempre orientando sobre as melhores condições para a realização do processo de soldagem.

Aos engenheiros mecânicos Kássio Maciel Kenitz, Jordano Fernandes Cordeiro e Frederico Allevalo Ramalho Filho, pelo conhecimento transmitido e pelo empréstimo de alguns componentes eletrônicos.

Ao aluno de graduação Roberto Almeida Gontijo, que tanto me ajudou nesta caminhada, mostrando-se sempre interessado em aprender e auxiliando de forma decisiva na realização dos testes com a utilização dos *softwares* MatLab e Mach 3.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela bolsa de mestrado e a Fundação Universitária Mendes Pimentel (Fump) pela alimentação de qualidade e de baixo custo.

Aos professores Alexandre Queiroz Bracarense do corpo docente do Departamento de Engenharia Mecânica da UFMG, Antônio Augusto Torres Maia do corpo docente do Departamento de Engenharia Mecânica da UFMG e Carlos Antônio Rennó do corpo docente do Departamento de Ciência da Computação do CEFET/MG, por aceitarem o convite de participar de minha banca examinadora.

A todos do Laboratório de Robótica, Soldagem e Simulação por compartilharem comigo conhecimentos e por me ajudarem com a parte experimental.

“Se cheguei onde cheguei, foi porque me apoiei em ombros de gigantes”.

Isaac Newton

“Só sabemos com exatidão quando sabemos pouco. À medida que vamos adquirindo conhecimentos, instala-se a dívida”.

Johann Goethe

“Os fenômenos ainda não foram suficientemente investigados. Quando o forem, poderemos depositar mais confiança na observação do que na especulação, e só confiaremos nesta na medida em que estiver de acordo com os fenômenos”.

Aristóteles

“Uma máquina pode fazer o trabalho de cinquenta pessoas comuns. Máquina alguma pode fazer o trabalho de um homem incomum.”

Elbert Hubbard

“Dedico este trabalho à minha esposa, uma pessoa tão especial.”

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

RESUMO

ABSTRACT

1. INTRODUÇÃO	20
1.1 Relevância do tema	21
1.2 Objetivos gerais.....	22
1.3 Objetivos específicos.....	22
1.4 Estrutura da dissertação	23
2. REVISÃO BIBLIOGRÁFICA	24
2.1 Sistemas de controle	24
2.1.1 Planta	24
2.1.2 Controladores.....	25
2.1.4 Sensores	31
2.2 Robótica industrial	32
2.2.1 Graus de liberdade.....	33
2.2.2 Graus de liberdade em robôs	35
2.2.3 Classificação dos robôs	35
2.2.4 Braço Robótico	40
2.3 Cinemática de Robôs Industriais.....	41
2.3.1 Descrição matricial para definição de posição e orientação	42
2.4 <i>Hardware</i> de acionamento.....	44
2.4.1 Fonte de Alimentação.....	44
2.4.2 Motores elétricos de corrente contínua (MCC)	45

2.4.3	<i>Drivers</i>	47
2.4.4	Sensores	49
2.4.5	Portas de interface paralela e <i>USB</i>	51
2.4.6	Gabinetes de controle de robôs industriais	52
2.6	<i>Retrofitting</i>	56
2.6.1	Realização de adequações a novas tecnologias em manipuladores robóticos industriais	57
2.6.2	Revitalização de máquinas obsoletas, com mau funcionamento e inexistência de peças de reposição	59
2.6.3	Automatização de máquinas de controle manual	60
2.7	Processo de soldagem a arco com arame tubular	62
2.7.1	Robotização do Processo <i>FCAW</i>	64
3.	METODOLOGIA	66
3.1	Projeto e construção do hardware	68
3.1.1	Seleção dos motores	68
3.1.2	Seleção dos sensores de realimentação	70
3.1.3	Seleção dos <i>drivers</i>	70
3.1.4	Projeto e confecção da fonte de alimentação	71
3.1.5	Construção do gabinete de controle	72
3.1.6	Montagem e testes dos motores, sensores de realimentação e sintonia dos controladores	74
3.1.7	Seleção da interface entre o computador e o <i>driver</i>	74
3.2	Seleção e programação do <i>software</i>	75
3.2.1	Execução do equacionamento geométrico direto e inverso do manipulador robótico (Cinemática Direta e Inversa do manipulador ASEA)	75
3.2.2	Seleção do <i>software</i> para geração de cinemática	85
3.2.3	Seleção do software para geração de pulsos	86
3.3	Realização de soldas através do processo de soldagem <i>FCAW</i>	88

3.4	Materiais e equipamentos	89
4.	RESULTADOS E DISCUSSÕES.....	94
4.1	Testes de falha do sistema	94
4.1.1	Dispositivos de segurança.....	95
4.2	Resposta a perturbações externas.....	98
4.3	Sintonia dos controladores PID.....	99
4.4	Limites de utilização de <i>hardware</i> e <i>software</i>	108
4.4.1	Fonte de alimentação.....	108
4.4.2	<i>Drivers</i> e <i>encoders</i>	108
4.4.3	Interface computador/ <i>driver</i>	109
4.4.4	Mach 3	109
4.4.5	Limites de velocidade do manipulador ASEA.....	110
4.5	Geração de trajetórias com os <i>softwares</i> selecionados.....	111
	Testes de posicionamento, velocidade e aceleração com carga dinâmica	115
4.6	Arrefecimento do gabinete de controle	116
4.7	Execução das soldas através do processo <i>FCAW</i>	116
4.8	Programa em MatLab para movimentação em qualquer orientação no espaço de trabalho	118
5.	CONSIDERAÇÕES FINAIS.....	121
6.	SUGESTÕES PARA TRABALHOS FUTUROS	122
7.	REFERÊNCIAS BIBLIOGRÁFICAS	123
7.1	Sítios visitados na internet.....	125
ANEXO A	126

LISTA DE FIGURAS

FIGURA 2.1 – Controle em malha fechada de um manipulador robótico.....	24
FIGURA 2.2 – Sistema de controle de posição de um motor elétrico.....	30
FIGURA 2.3 – Sistema de controle de cinemática de um motor, a partir de chaveamento de um transistor.....	31
FIGURA 2.4 – Robôs antropomorfos para soldagem de carroceria de automóvel.....	33
FIGURA 2.5 – Objeto sobre a mesa com três graus de liberdade.....	34
FIGURA 2.6 – Graus de liberdade utilizados para orientação da tocha na soldagem GMAW: (a) movimento de puxar e empurrar o arco elétrico e (b) ângulo em relação ao chanfro.....	34
FIGURA 2.7 – Juntas do braço humano totalizando sete graus de liberdade.....	35
FIGURA 2.8 – (a) Robô de Coordenadas Cartesianas e (b) Movimentos de cada eixo.....	36
FIGURA 2.9 – (a) Robô de Coordenadas Cilíndricas e (b) seu espaço de trabalho.....	37
FIGURA 2.10 – (a) Robô de Coordenadas Polares ou Esféricas e (b) movimentos de cada eixo.....	37
FIGURA 2.11 – (a) Robô PUMA 500 e (b) Movimentos de cada junta de um Robô de Coordenada de Revolução ou Articulado.....	38
FIGURA 2.12 – (a) Robô SCARA modelo TH1050 produzido pela Toshiba e (b) Movimentos de cada junta de um robô SCARA.....	39
FIGURA 2.13 – Simulador de dirigibilidade de veículo.....	39
FIGURA 2.14 – (a) Robô Antropomorfo fabricado pela KUKA e (b) As articulações do braço humano e sua relação com as articulações projetadas nas juntas dos Robôs Antropomorfos.....	40
FIGURA 2.15 – Robô PUMA 560.....	41

FIGURA 2.16 – Espaços utilizados para definir o posicionamento de um manipulador robótico industrial.....	42
FIGURA 2.17 – Localização de um objeto em posição e orientação.....	43
FIGURA 2.18 – Diagrama em blocos de uma fonte de alimentação.....	45
FIGURA 2.19 – Célula básica de um MCC com uma espira.....	46
FIGURA 2.20 – Motor elétrico de corrente contínua com seus principais componentes.....	46
FIGURA 2.21 – (a) Partes internas de um motor de corrente contínua e (b) representação esquemática.....	47
FIGURA 2.22 – <i>Servodriver</i> modelo G320 fabricado pela Geckodrive.....	48
FIGURA 2.23 – (a) Sinal de controle de um motor com ciclo de trabalho de 30% e (b) rotação desenvolvida pelo motor.....	49
FIGURA 2.24 – (a) Vista explodida de um <i>encoder</i> e (b) detector óptico e disco graduado.....	50
FIGURA 2.25 – (a) <i>Encoder</i> fabricado pela USDigital, (b) módulo de transmissão do <i>encoder</i> óptico e (c) disco de <i>encoder</i>	51
FIGURA 2.26 – (a) Porta paralela e (b) porta <i>USB</i>	52
FIGURA 2.27 – Gabinete de controle do robô modelo IRB 2000 fabricado pela ABB.....	53
FIGURA 2.28 – Gabinete de controle com interface paralela fabricado no LRSS.....	54
FIGURA 2.29 – Componentes básicos de uma máquina CNC.....	55
FIGURA 2.30 – (a) Novo painel de controle utilizado na atualização de <i>hardware</i> de um robô IRB 2000 e (b) painel de controle original.....	58
FIGURA 2.31 – Diagrama em blocos da arquitetura de controle desenvolvida.....	58
FIGURA 2.32 – Máquina IDAB <i>Marathoner Stacker</i>	59
FIGURA 2.33 – Diagrama da arquitetura utilizada no <i>retrofitting</i>	60

FIGURA 2.34 – (a) Gabinete de controle utilizado no <i>retrofitting</i> e (b) torno mecânico convencional controlado por CNC.....	61
FIGURA 2.35 – Soldagem <i>FCAW</i> (esquemática).....	63
FIGURA 2.36 – Equipamento para soldagem <i>FCAW</i>	63
FIGURA 2.37 – Célula robótica de soldagem do LRSS.....	65
FIGURA 3.1 – Fluxograma com a metodologia proposta para a realização da remanufatura.....	67
FIGURA 3.2 – (a) Motor de corrente contínua substituído e (b) motor substituto no processo de <i>overhauling</i>	69
FIGURA 3.3 – Testes do motor operando em malha aberta.....	69
FIGURA 3.4 – <i>Encoder</i> fabricado pela USDigital com resolução de 8.192 pulsos por revolução, quando operando em quadratura.....	70
FIGURA 3.5 – <i>Servodrive</i> G320 utilizado nos experimentos.....	71
FIGURA 3.6 – Fonte de alimentação de 1.750W utilizada nos experimentos.....	72
FIGURA 3.7 – Arquitetura de montagem do gabinete de controle.....	72
FIGURA 3.8 – Arranjo de ligação dos <i>drivers</i> G320, (a) motores 1, 2 e 3 e (b) motores 4 e 5.....	74
FIGURA 3.9 – Arranjo de ligação da placa modelo C1G – Breakout Board.....	74
FIGURA 3.10 – Placa com interface paralela modelo C1G-Breakout Board.....	75
FIGURA 3.11 – Graus de liberdade do manipulador ASEA.....	76
FIGURA 3.12 – Detalhe de acionamento da segunda junta do manipulador ASEA.....	81
FIGURA 3.13 – Detalhe de acionamento da terceira junta do manipulador ASEA.....	82
FIGURA 3.14 – Sistema de transmissão por hastes para acionamento da junta 4 do manipulador ASEA.....	83

FIGURA 3.15 – Tela de comando do <i>software</i> Mach 3 para operar duas fresas simultaneamente.....	87
FIGURA 3.16 – Diagrama esquemático da malha de controle de um motor.....	87
FIGURA 4.1 – Testes iniciais em bancada.....	94
FIGURA 4.2 – Vista de cima do gabinete de controle utilizado nos experimentos, com destaque para o disjuntor e o botão vermelho de emergência.....	96
FIGURA 4.3 – Esquema elétrico das ligações de hardware do gabinete de controle utilizado nos experimentos.....	97
FIGURA 4.4 – Testes de perturbações externas com carga estática de 6Kg.....	98
FIGURA 4.5 – Potenciometro multivoltas conectado ao eixo do motor 5.....	100
FIGURA 4.6 – Resposta em malha aberta do motor 4 para AS configurações: (a) robô retraído, (b) robô estendido e (c) resposta ao degrau de 2V.....	102
FIGURA 4.7 – <i>Driver</i> evidenciando os três potenciômetros das ações de controle PID.....	102
FIGURA 4.8 – Sinal de entrada gerado pela placa PIC (canal 1) e resposta do motor verificada através da queda de tensão no potenciometro (canal 2) para uma sintonia 3/0/0.....	103
FIGURA 4.9 – Sinal de entrada gerado pela placa PIC (canal 1) e resposta do motor verificada através da queda de tensão no potenciometro (canal 2) para uma sintonia 3/0/1.....	104
FIGURA 4.10 – Sinal de entrada gerado pela placa PIC (canal 1) e resposta do motor verificada através da queda de tensão no potenciometro (canal 2) para uma sintonia 3/2/1.....	105
FIGURA 4.11 – Resposta do motor 5 para uma entrada em rampa, com diferentes sintonias do controlador PID.....	106
FIGURA 4.12 – Comparação da resposta em malha fechada para o motor 5, com sintonia 3/2/1, para diferentes configurações das juntas: (a) resposta do motor na configuração estendida com carga de 6kg e (b) resposta do motor na configuração zero de referência sem carga.....	107
FIGURA 4.13 – Código G gerado através das cinemáticas inversas e do programa de geração de trajetória.....	114

FIGURA 4.14 – Marcação dos pontos de referência para verificação de posicionamento.....	115
FIGURA 4.15 – Cordão de solda obtido no Experimento 1.....	116
FIGURA 4.16 – Cordão de solda obtido no Experimento 5.....	117
FIGURA 4.17 – (a) Vista de cima da solda em chanfro obtida no Experimento 8 e (b) macrografia, nital 5%.....	118
FIGURA 4.18 – Célula robótica do LRSS constituída pelo manipulador ASEA, após o processo de <i>overhauling</i>	120

LISTA DE TABELAS

TABELA 2.1 – Ação de controle Proporcional, Integral e Derivativa com as respectivas equações, respostas típicas a uma entrada degrau e implementação em circuitos com amplificadores operacionais.....	28
TABELA 2.2 – Relação entre as grandezas físicas e os sensores utilizados em suas medições.....	32
TABELA 3.1 – Descrição da função de cada pino do <i>driver</i> Geckodrive G 320.....	73
TABELA 3.2 – Ligações do <i>encoder</i> do motor ao <i>driver</i> G320.....	73
TABELA 3.3 – Parâmetros de Denavit-Hartenberg para o manipulador ASEA.....	77
TABELA 3.4 – Dimensões do manipulador ASEA utilizados para o cálculo da cinemática inversa de atuadores.....	80
TABELA 3.5 – Parâmetros de soldagem para a realização da parte experimental.....	88
TABELA 3.6 – Lista de materiais – <i>kit-overhauling</i> do manipulador ASEA.....	89
TABELA 3.7 – Lista de <i>softwares</i> utilizados no projeto de <i>overhauling</i> do manipulador ASEA.....	90
TABELA 3.8 – Lista de equipamentos utilizados no projeto de <i>overhauling</i> do manipulador ASEA.....	91
TABELA 3.9 – Lista de instrumentos de medição utilizados no projeto de <i>overhauling</i> do manipulador ASEA.....	92
TABELA 3.10 – Lista de matérias e consumíveis utilizados nos experimentos do projeto de <i>overhauling</i> do manipulador ASEA.....	93
TABELA 4.1 – Ajuste dos parâmetros do controlador PID e erro residual para cada motor.....	108
TABELA 4.2 – Resposta do manipulador ASEA para diferentes valores de aceleração.....	115

LISTA DE ABREVIATURAS E SIGLAS

A – Corrente em Amperes
ABB – Asea Brown Boveri
C.A. – Corrente Alternada
C.C. – Corrente Contínua
CAD – *Computer-aided design* (Projeto Assistido por Computador)
CAD/CAM – Integração entre CAD e CAM
CAM – *Computer-aided manufacturing* (Manufatura assistida por computador)
CAN-Bus – Barramento de dados
CLP – Controlador Lógico Programável
CNC – Controle Numérico de Comando de máquinas ferramentas
Controlador PID – Controlador Proporcional, Integral e Derivativo
Conversor A/D – Conversor Analógico/Digital
Conversor D/A – Conversor Digital/Analógico
DBCP – Distância Bico de Contato Peça
FCAW – *Flux-Cored Arc Welding* (soldagem a arco com arame tubular)
G code – Código G
GBps – GigaBitz por segundo
IHM – Interface Homem-Máquina
ISO – *International Organization for Standarization* (Organização Internacional para Padronização)
kg – quilograma
kHz – quiloHertz
LCD – *Liquid Crystal Display* (Display de Cristal Líquido)
LED – *Light Emitting Diode* (Diodo Emissor de Luz)
Line Printer (DB-25) – Conector utilizado em linha de impressão
LRSS – Laboratório de Robótica Soldagem e Simulação
mA – Corrente em miliAmperes
MatLab – *Matrix Laboratory* (Laboratório de Matrizes)
MBps – MegaBitz por segundo
MCC – Motor de corrente contínua
mm – Comprimento em milímetros

mm/s – Velocidade em milímetros por segundo
mm/s² – Aceleração em milímetros por segundo ao quadrado
ModBus - Barramento Modular
ms – Tempo em milissegundos
mW – Potência em miliWatts
N_p – Número de espiras no primário do transformador
N_s – Número de espiras no secundário do transformador
Parâmetros D-H – Parâmetros de Denavit-Hartenberg
Placas de I/O – Placas de *Input* e *Output* (entrada e saída)
PUMA – *Programmable Universal Machine for Assembly* (Máquina Universal Programada para a Montagem)
PWM – *Pulse With Modulation* (Modulação por Largura de Pulso)
RF – *Radio Frequency* (Frequência de Rádio)
Robot script – Código fonte da linguagem de programação
RPM – Rotação por minuto
SCARA – *Selective Compliant Articulated Robot for Assembly* (Robô Articulado para Operações de Montagem)
seg – Tempo em segundos
TCP – *Tool Center Point* (Ponto Central da Ferramenta)
Trafo – Transformador
USB – *Universal Serial Bus* (Barramento Serial Universal)
V – Tensão em Volts
V_{dc} – *Volts direct current* (Volts em corrente contínua)
V_p – Tensão no primário do transformador
V_s – Tensão no secundário do transformador
W – Potência em Watts

RESUMO

Atualmente, é cada vez maior o número de manipuladores robóticos que são descartados no chão de fábrica e são classificados como equipamentos inservíveis, apesar de ainda estarem em condições de serem reutilizados. Este problema ocorre devido ao fato dos fabricantes de robôs utilizarem uma arquitetura de controle totalmente fechada para seus sistemas de *hardware* e *software*. Desta forma, não há intercambialidade entre os gabinetes de controle e as linguagens de programação utilizadas para geração de comandos, o que encarece a execução de reparos, modificações e implementações ao sistema, pois estes processos se tornam dependentes de cada fabricante. Nesse contexto, este trabalho propõe uma nova metodologia para a execução da remanufatura de manipuladores robóticos industriais. Nesta nova metodologia, é proposto um novo conceito denominado *overhauling*, onde apenas a estrutura mecânica do robô é reaproveitada, tornando o processo de remanufatura mais rápido e de maior confiabilidade, quando comparado a outros processos como o *retrofitting*. O processo de *overhauling* propõe uma arquitetura de controle totalmente aberta. Assim, os dispositivos de *hardware* são intercambiáveis com outros processos e a linguagem de programação utilizada pode ser substituída.

Para validar a metodologia proposta, foi realizado o *overhauling* do manipulador robótico ASEA IRB6-S2. Desta forma, a remanufatura mostrou-se uma solução bastante viável para a reciclagem de robôs, pois o manipulador IRB6-S2 apresentou boa precisão e repetibilidade a um custo bastante reduzido quando comparado ao de um robô novo com características similares.

ABSTRACT

Currently, an increasing number of robotic manipulators are discarded on factory floor because they are classified as unserviceable equipment, while still being able to be re-used. This problem occurs due to the fact that robot manufacturers use fully closed architecture control to their hardware and software systems. Thus, there is no interchangeability between the control cabinets and the programming languages used to generate kinematics, which make the repairs, modifications and implementations to system more expensive, due to these processes dependent of each manufacturer. In this context, this work proposes a new methodology to the remanufacturing of industrial robotic manipulators. This new methodology proposes a new concept called overhauling, in which only the mechanical structure of the robot is re-used, making the remanufacturing process faster and more reliable when compared to other processes such as retrofitting. The overhauling process proposes a fully open architecture control. Thus, the hardware devices are interchangeable with other processes and the programming language can be substituted.

To validate this new methodology proposed, the ASEA IRB6-S2 robot overhauling was performed. This way, the remanufacturing proved to be a very feasible solution for the recycling of robots, because the manipulator IRB6-S2 showed good accuracy and repeatability at a greatly reduced cost compared to a new robot with similar characteristics.

1. INTRODUÇÃO

O Laboratório de Robótica, Soldagem e Simulação da Universidade Federal de Minas Gerais (LRSS/UFMG) recebeu a doação em 1998 de um robô fabricado pela ASEA (atual ABB), no ano de 1977. O manipulador robótico é o modelo ASEA IRB6-S2 (manipulador ASEA), de cinco graus de liberdade e capacidade de carga de seis quilos, construído para aplicações de soldagem na indústria automobilística. Este robô apresentava sua estrutura mecânica em boas condições, mas seu sistema de controle estava com seus componentes eletrônicos e eletromecânicos em péssimas condições de uso, além de obsoletos.

O fato do LRSS trabalhar em projetos para a construção de pequenos robôs didáticos de soldagem e robôs industriais para aplicações especiais motivou a execução da remanufatura do robô para aplicações de soldagem. A proposta inicial foi executar o *retrofitting* do manipulador robótico. Nesta filosofia, os componentes eletromecânicos e eletrônicos do gabinete de controle original são reaproveitados, fazendo adaptações de forma a tornar o gabinete intercambiável com qualquer fabricante de manipuladores robóticos.

Atualmente, cada fabricante de manipulador robótico possui sua linguagem de programação e gabinete de controle para aplicações em motores e sensores de realimentação específicos, o que torna a intercambialidade, a programação e a manutenção processos difíceis de serem realizados. Cada fabricante possui sua programação *robot script*, de linguagem própria e comandos específicos, o que exige a realização de cursos específicos para a habilitação de profissionais que executem mudanças na cinemática e realizem manutenção no gabinete de controle.

Tendo em vista o grande crescimento tecnológico e o evidente aumento da concorrência no mercado da área de robótica, se torna de extrema importância o desenvolvimento de produtos padronizados, que funcionem com todas as marcas disponíveis. Com este intuito, durante a realização do *retrofitting*, optou-se pela construção de um gabinete de controle com interface paralela *Line Printer* com conector de 25 pinos (DB-25), onde a trajetória do robô pode ser criada em linguagens de programação universais como MatLab, C++, Java, Pascal, Delphi, dentre outras. O gabinete de controle poderia ser facilmente reutilizado em outros manipuladores robóticos ou até mesmo em outras máquinas como fresas e tornos CNC, desde que fossem feitos pequenos ajustes na fonte de alimentação de forma a atender aos motores específicos de cada processo.

Durante a execução do *retrofitting*, observou-se que o reaproveitamento de componentes usados seria uma prática difícil de ser executada no caso específico do manipulador ASEA, devido ao seu tempo prolongado de utilização (35 anos) e devido ao fato da difícil adaptação de tecnologia obsoleta às novas tecnologias. Desta forma, chegou-se à conclusão que seria mais viável a troca de todos os componentes eletrônicos e eletromecânicos reutilizando apenas a mecânica do robô. Com esta mudança de metodologia, o processo de *retrofitting* é substituído por um novo conceito denominado de *overhauling*.

Nesta nova filosofia de trabalho, apenas a parte mecânica do manipulador ASEA seria reaproveitada. Para a parte de *software* e *hardware* (motores, sensores de posição, gabinete de controle e *softwares* geradores de trajetória), haveria a criação de um conjunto de componentes denominado “*Kit Overhauling*”, onde sua utilização seria interessante em situações em que os manipuladores robóticos têm idade avançada, o que torna difícil a adequação de novas tecnologias e a reutilização dos componentes que não têm nenhum histórico de operação, como é o caso do manipulador ASEA. Com esta nova metodologia, há economia de tempo e custos devido ao fato de não ser necessário o teste de componentes antigos e muitas vezes obsoletos e o processo torna-se de maior confiabilidade, pois, com a utilização de componentes novos, pode-se dar garantias do tempo de uso e da necessidade de manutenção.

1.1 Relevância do tema

Atualmente, é grande o número de manipuladores robóticos que estão sendo inutilizados por grandes indústrias, principalmente do ramo automobilístico. Estes robôs, geralmente, são sucateados devido a sua tecnologia ultrapassada e ao custo elevado para a execução de reparos. Neste contexto, a remanufatura passa a se tornar uma alternativa viável para a reciclagem de robôs inservíveis.

Outra questão importante a ser considerada é o fato de manipuladores robóticos disponíveis em meios acadêmicos serem, geralmente, de fabricantes que possuem uma arquitetura fechada e uma linguagem de programação própria. Desta forma, os trabalhos ficam restritos aos limites impostos por cada fabricante. A criação de um manipulador com arquitetura e linguagem de programação abertas faz com que esses limites sejam quebrados e o pesquisador tenha acesso à tecnologia aplicada em todos os sistemas de *software* e *hardware*.

1.2 Objetivos gerais

O objetivo geral do trabalho é executar a remanufatura do manipulador ASEA reutilizando apenas sua estrutura mecânica. O trabalho de reconstrução dos sistemas de controle e acionamento do manipulador robótico é dividido em duas etapas, que consistem no projeto, especificação e montagem de *hardware*, na qual é proposto a montagem de um gabinete de controle com arquitetura aberta, e especificação e programação de *software*, na qual optou-se por uma linguagem de programação não proprietária.

1.3 Objetivos específicos

Os objetivos específicos na realização da remanufatura do manipulador ASEA serão os seguintes:

- cálculo e especificação dos motores;
- especificação dos sensores de realimentação;
- definição da melhor estrutura mecânica para o gabinete de controle;
- especificação e construção de componentes eletrônicos da fonte de alimentação;
- especificação de atuadores (*drivers*);
- montagem dos subsistemas de *hardware*:
 - motores;
 - sensores;
 - fonte de alimentação;
 - atuadores;
 - interface computador/atuadores;
- ajuste do controlador PID, calibrando as ações de comando proporcional, integral e derivativa;
- especificação do *software* com linguagem de programação aberta;
- programação de trajetórias, utilizando o MatLab;
- execução de testes, verificando posicionamento, precisão e repetibilidade;
- execução de testes, com carga de seis quilos, verificando posicionamento, precisão e repetibilidade;
- execução de cordões de solda, utilizando o processo *FCAW*, de forma a validar o processo de remanufatura;
- levantamento de resultados e discussões, observando o custo-benefício do processo de remanufatura.

1.4 Estrutura da dissertação

Esta dissertação está estruturada em sete capítulos. O capítulo 2 apresenta a revisão bibliográfica necessária ao entendimento dos assuntos abordados. No capítulo 3, é apresentada a metodologia adotada para a execução do processo de remanufatura do manipulador ASEA, juntamente com os materiais, equipamentos e instrumentos utilizados. No capítulo 4, são apresentados os resultados obtidos, bem como os passos que devem ser seguidos para a construção e teste dos sistemas de controle. No capítulo 5, é feita uma análise que retoma os principais pontos abordados durante o trabalho e as conclusões obtidas a partir do estudo realizado. No capítulo 6, são feitas sugestões para trabalhos futuros. O capítulo 7 contém as referências bibliográficas que foram utilizadas para a realização do trabalho.

2. REVISÃO BIBLIOGRÁFICA

2.1 Sistemas de controle

Um sistema de controle é composto basicamente de quatro elementos: planta, atuador, sensor e controlador (OGATA, 2003). Para o caso particular do manipulador robótico, a planta seria o braço robótico, o atuador é representado pela placa controladora ou *driver*, juntamente com os motores, o sensor seria o *encoder* e o controlador do tipo PID. A Figura 2.1 apresenta um diagrama esquemático de controle do acionamento de uma junta de um manipulador robótico. As variáveis deslocamento, velocidade e aceleração são mensuradas através de um *encoder*, que é um conversor Analógico/Digital. O sinal digital obtido é fornecido ao controlador, onde é comparado com uma trajetória programada de referência e, se houver alguma divergência (erro), o controlador envia um sinal ao atuador, fazendo com que a trajetória siga o valor desejado.

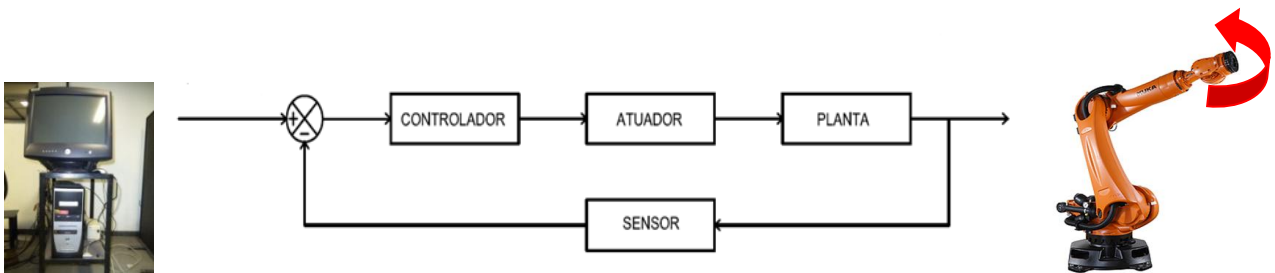


FIGURA 2.1 - Controle em malha fechada de um motor de um manipulador robótico.

2.1.1 Planta

Uma planta consiste no corpo do processo, isto é, um sistema a controlar que pode ser parte de um equipamento ou apenas um conjunto de componentes de um equipamento que funcione de maneira integrada, com o objetivo de realizar determinada operação. Em aplicações de engenharia, denomina-se sistema a controlar qualquer objeto físico a ser controlado, como um componente mecânico, um forno, um reator químico, um motor de combustão interna ou uma espaçonave. Para o caso específico dos robôs manipuladores, a planta consiste no braço robótico, parte física do sistema que desempenhará as tarefas a serem executadas.

Sistemas de controle com realimentação

Com a necessidade de atender aos padrões e normas cada vez mais exigentes e melhoria do produto, o setor industrial passou a investir em servomecanismos, a fim de buscar maior qualidade e controle dimensional. A nomenclatura servo deve-se ao fato desses mecanismos

terem modificações em parâmetros do controlador, geralmente ganho de amplificadores, dependendo do sinal de erro emitido por um sensor.

A realimentação consiste em instalar na planta um sensor que envie um sinal a ser comparado com o sinal de entrada (*set point*). Como exemplo, pode-se citar um termopar (sensor) em um forno (planta), gerando um sinal de erro que poderá ser zerado através de controladores PI (Proporcional-Integral). Quando se utiliza a realimentação, a planta é considerada um sistema em malha fechada.

2.1.2 Controladores

A finalidade do controlador é comparar a saída efetiva da planta com o comando de entrada, proporcionando um sinal de controle que reduza o erro a valores próximos de zero e controle parâmetros do processo. Esses parâmetros podem ser o tempo de subida, máximo sobre-sinal ou tempo de acomodação (GROOVER, 1989).

O controlador geralmente consiste em um ponto de soma, onde os sinais de entrada e saída são comparados, gerando um sinal de erro que deve tender a zero pela ação de controle. O atuador é usado na robótica para converter a ação de controle em movimento físico do manipulador. O controlador e o atuador podem ser operados por meios pneumáticos, hidráulicos, mecânicos ou eletrônicos, ou combinações destes.

Em servosistemas, são geralmente utilizados seis tipos de controladores:

1. Liga-desliga;
2. Proporcional;
3. Integral;
4. Proporcional-integral (PI);
5. Proporcional-derivativo (PD);
6. Proporcional-integral-derivativo (PID).

Nos controladores liga-desliga, o elemento de controle fornece dois níveis de controle, ligado ou desligado. Um exemplo de utilização deste tipo de controlador é o termostato utilizado em equipamentos de ar condicionado. O controlador proporcional é uma ação de controle mais suave que a liga-desliga, pois atua como um amplificador com um ganho denominado K_p . Esse controlador, contudo, não garante erro zero em regime permanente. Uma forma de zerar o erro em regime permanente é utilizar um controlador integral, que devido a sua resposta lenta, é utilizado em conjunto com o controlador proporcional, gerando a ação de controle PI. Apesar do

controlador PI zerar o erro em regime permanente, houve a necessidade de controlar outros parâmetros para que o sistema forneça resposta rápida e bom controle de estabilidade, possibilitado através da ação de controle PID.

Nos próximos itens, serão realizadas análises das funções das ações de controle proporcional, integral e derivativa, de forma a elucidar a influência de cada ação no desempenho de sistemas em malha fechada.

Ação de Controle Proporcional (P)

O circuito eletrônico equivalente de um Controlador Proporcional é composto por amplificadores operacionais, onde o valor do ganho proporcional (K_p) é ajustado de acordo com os valores das resistências conectadas ao amplificador operacional, como pode ser verificado na Tabela 2.1. Quando há a necessidade de tornar a resposta do sistema mais rápida e oferecer maior rigidez, o valor do ganho de K_p deve ser incrementado de forma a aumentar o ganho do amplificador. O problema nessa situação é que o aumento de K_p é acompanhado de um aumento do erro sistemático (erro residual ou erro estacionário) e aumento da amplitude nas oscilações (maior sobre-sinal na resposta). Ambos os efeitos crescem à medida que o ganho K_p é incrementado. Para atenuar, ou até mesmo cancelar os efeitos de erro estacionário e oscilações, foram criadas as ações de controle integral e derivativa.

Ação de Controle Integral (I)

Como pode ser visto na Tabela 2.1, a Ação de Controle Integral acrescenta um integrador à equação de malha fechada do sistema. Sempre que a ação P for um sinal de controle não nulo, haverá um erro residual (soma entre o *set point* e o sinal realimentado). Desta forma, o integrador tem a finalidade de zerar o erro estacionário, realizando o somatório (integração) da área sob a curva do sinal de erro atuante. Assim, a cada instante que ocorre a realimentação da malha, ocorre concomitantemente um decréscimo do erro residual.

Em sistemas de controle de posição, se for programado para um motor realizar um deslocamento de X metros e o mesmo deslocar $1,1X$ metros, o sensor realimentará a malha com $1,1X$ metros. No controlador, esse sinal será somado ao sinal de referência (*set point*) e um sinal de erro de $-0,1X$ metros será gerado (realimentação negativa). Desta forma, a ação I armazena esse erro, atuando na resposta do sistema decrementando $0,1X$ metros de $1,1X$ metros e zerando o erro residual.

Sempre que uma ação I for sintonizada, deve-se tomar cuidado para não aumentar demasiadamente K_i (constante integrativa), o que pode conduzir o sistema a uma resposta oscilatória indesejável, podendo levar a planta a uma instabilidade catastrófica (OGATA, 2003).

Ação de Controle Derivativa (D)

A ação de controle derivativa (D) responde a uma taxa de variação do erro atuante e pode produzir uma correção significativa antes que o valor do erro atuante se torne muito elevado (OGATA, 2003). O controle derivativo tende a aumentar a estabilidade do sistema, aumentando o amortecimento do mesmo, pois reduz as oscilações causadas por variações do erro.

As desvantagens que a ação de controle derivativa possui são que quando ela atua sobre uma planta que gera grande quantidade de ruído, a ação acaba derivando o ruído e o controlador entende que está atuando em um erro de proporções elevadas. A outra desvantagem é que essa ação de controle atua na variação do erro. Desta forma, a ação derivativa interpreta o erro constante como se não houvesse erro no sistema.

TABELA 2.1 – Ação de controle Proporcional, Integral e Derivativa com as respectivas equações, respostas típicas a uma entrada degrau e implementação em circuitos com amplificadores operacionais.

	Ação de controle	Equações	Resposta típica a uma entrada em degrau	Circuitos com amplificadores operacionais
1	P	$k_p \cdot e(t)$		
2	PD	$k_p \cdot e(t) + k_d \frac{de(t)}{dt}$		
3	PI	$k_p \cdot e(t) + k_i \int e(t) dt$		
4	PID	$k_p \cdot e(t) + k_i \int e(t) dt + k_d \frac{de(t)}{dt}$		

O tipo de controlador a ser utilizado (tipo P, PI, PD ou PID), depende da planta a ser controlada e da resposta desejada para determinada aplicação. Em sistemas térmicos, geralmente, são utilizados controladores do tipo PI, pois a inércia do sistema é elevada, o que torna a resposta da planta lenta, quando comparada a plantas que envolvem controle de velocidade de motores. Com esta resposta lenta, o controlador atua no sistema antes que um sobre-sinal elevado seja atingido. Em sistemas que exigem precisão de posicionamento, manipuladores robóticos, por exemplo, são utilizados controladores do tipo PID, pois a resposta do motor a um *set point* é muito rápida, o que geralmente causa uma elevada oscilação.

Ao iniciar a sintonia de um controlador de uma planta, não é muitas vezes vantajoso iniciar o processo com um controlador PID durante a primeira tentativa, pois um controlador do tipo P

pode fornecer uma resposta satisfatória para uma determinada aplicação. Em controladores do tipo PI, PD e PID, cuidados adicionais devem ser adotados, pois o aumento de K_i e K_d (constante derivativa) nem sempre são benéficos ao sistema. Quando ambos os ganhos são incrementados para valores demasiadamente elevados, a estabilidade do sistema pode ser prejudicada de forma a levar a planta a uma falha catastrófica. Isto significa que o sistema passa a oscilar em uma amplitude elevada, o que pode levar a estrutura da planta operar no regime plástico, tendo por consequência a ruptura de sua estrutura física.

Métodos de ajuste de Controladores PID

À partir da década de 40, iniciou-se a corrida para a criação de métodos para a sintonia de controladores PID. Um dos primeiros métodos presentes na literatura foi o Método ZN desenvolvido por J. G. Ziegler e N. B. Nichols, sendo o primeiro método de ajuste sistemático dos parâmetros de um controlador PID. Nesse método, Ziegler e Nichols propuseram regras para a determinação de valores do ganho proporcional K_p , do tempo integral T_i e do tempo derivativo T_d , baseadas nas características da resposta temporal de uma planta. Após esse trabalho, numerosas regras de sintonia para controladores PID vêm sendo propostas. Essas regras de sintonia estão disponíveis na literatura e geralmente são aplicadas a fabricantes e processos específicos (OGATA, 2003). Esses métodos, geralmente, prevêm o levantamento dos parâmetros K_p , K_i e K_d para sistemas que possuem uma equação de transferência linear.

Em plantas industriais, nem sempre a equação de transferência que modela um sistema possui um caráter invariante no tempo. Geralmente, esses sistemas têm mudanças nessa equação principalmente devido a efeitos inerciais. Por exemplo, quando um ar condicionado é utilizado para climatização de uma sala de aula, ocorre uma variação da equação de transferência que rege o sistema à medida que o número de alunos varia nessa sala. Esse efeito ocorre basicamente devido a variações na inércia térmica, devido a flutuações na carga térmica da sala. Outro exemplo é o que ocorre com manipuladores robóticos industriais, onde mudanças no valor da carga no TCP (*Tool Center Point*), concomitante a variações na configuração dos ângulos das juntas, fazem com que ocorram variações na equação de transferência do sistema. Para esses casos, a literatura prevê controladores adaptativos que possuem algoritmos que refazem a sintonia das ações de controle em intervalos de tempo que decrescem à medida que o sistema necessita de uma maior precisão. Existem vários tipos de controles adaptativos, dentre eles pode-se citar o controle adaptativo por modo de referência, o controle adaptativo com reguladores auto-ajustáveis e o controle adaptativo com planejamento de ganho (MAIA, 2005).

2.1.3 Atuadores

O atuador é um dispositivo que atua sobre a variável manipulada de modo a afetar a variável controlada. Controlar significa medir o valor da variável controlada do sistema e utilizar a variável manipulada do sistema, através do atuador, para corrigir ou limitar os desvios do valor medido a partir de um valor desejado (OGATA, 2003).

Para exemplificar, é dado o exemplo de um sistema de controle de posição de um motor elétrico (FIG. 2.2). O deslocamento angular do motor é medido por um encoder, que é um dispositivo digital. O sinal digital obtido é fornecido ao controlador por meio de uma interface. Esse sinal digital é comparado com um sinal de referência na entrada do controlador e, se houver alguma divergência (erro), o controlador envia um sinal ao *driver*, fazendo com que o deslocamento angular do motor atinja o valor desejado.

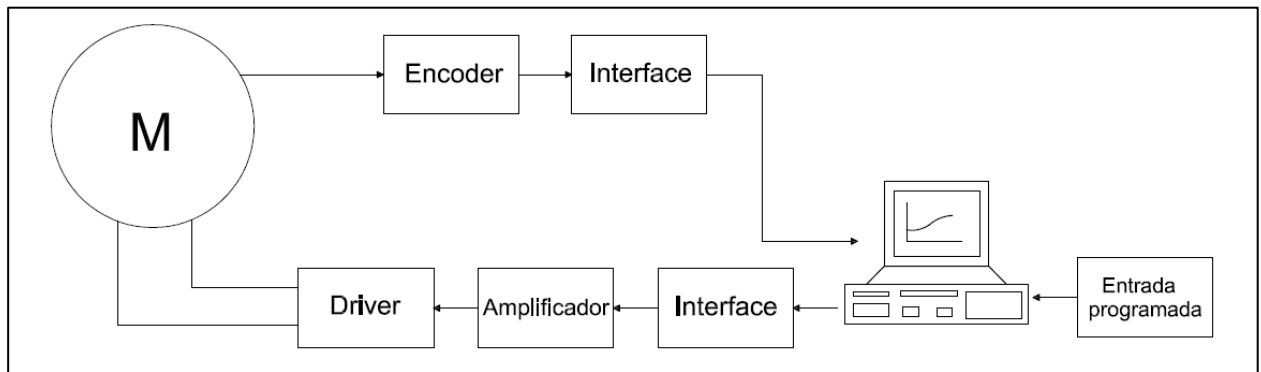


FIGURA 2.2 - Sistema de controle de posição de um motor elétrico.

ADAPTADO DE: OGATA, 2003.

Os sinais transmitidos por placas de *I/O* de computadores, controladores lógicos programáveis e placas de comunicação, utilizados em sistemas de controle, são sinais de baixa potência. Geralmente, estes sinais são de 4 a 20mA, de 0 a 10V e ou são pulsos de 5V e alguns mW. Conseqüentemente, os sinais oriundos de sistemas de controle não possuem potência suficiente para realizar o acionamento de uma planta. Desta forma, a saída destes controladores deve ser ligada a amplificadores operacionais ou a outros dispositivos que funcionem como transdutores. Estes amplificadores e transdutores são conhecidos como atuadores.

Para o caso específico dos manipuladores robóticos industriais, onde o acionamento das juntas é feito através de servomotores, os atuadores funcionam como amplificadores de potência, pois a partir de um sinal de pequena amplitude e potência, oriundo de um computador ou

microcontrolador, pode-se controlar um sinal de potência suficiente para acionar os motores, como pode ser verificado na Figura 2.3. Nestas aplicações, o atuador, também conhecido como *driver*, tem seu funcionamento similar ao de um relé com capacidade de acionamento muito rápida (microsegundos).

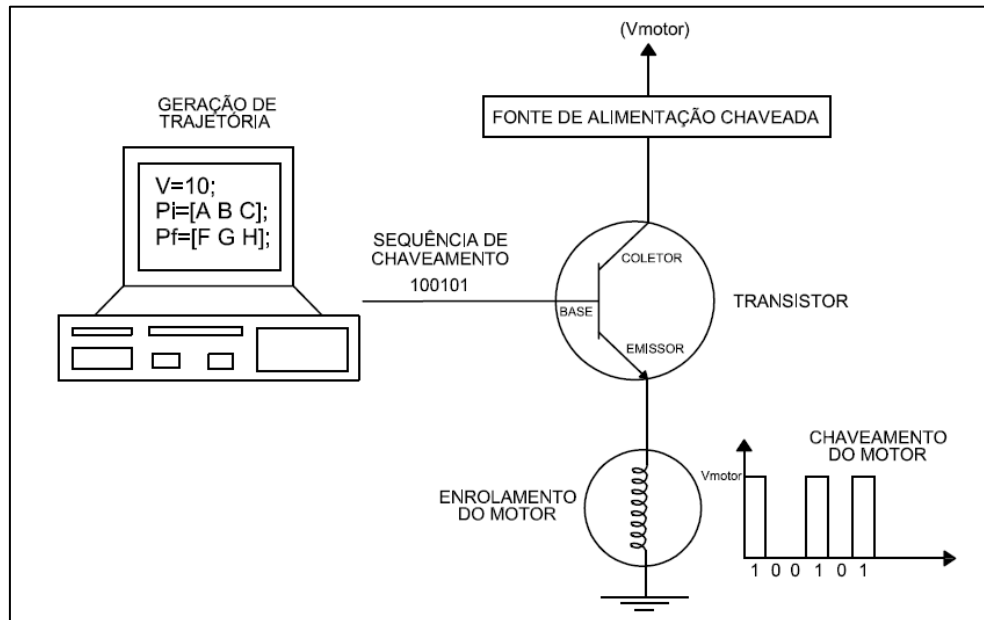


FIGURA 2.3 - Sistema de controle de cinemática de um motor, a partir de chaveamento de um transistor.

2.1.4 Sensores

Os sensores são dispositivos utilizados quando se deseja mensurar uma variável controlada em um determinado processo. Em sua concepção mais clássica, os sensores são considerados transdutores que transformam o valor da variável que se deseja medir em um valor inteligível pelo operador. Em sistemas de controle em malha fechada, o sensor é o elemento que fecha a malha, tornando o controle mais preciso e menos vulnerável a perturbações externas.

O sinal emitido pelos sensores geralmente são analógicos, sendo necessária a utilização de conversores A/D (Analógico-Digital) para que o controlador entenda qual é o valor da variável que está sendo mensurada no processo. O sinal digital obtido é fornecido ao controlador, sendo comparado com um sinal programado de referência. Se houver alguma divergência (erro), o controlador envia um sinal ao atuador, fazendo com que a variável controlada atinja o valor desejado.

Na Tabela 2.2, há uma relação entre as grandezas físicas e os sensores utilizados em suas medições. Um sensor muito utilizado em aplicações de robótica é o *encoder*, pois este sensor pode medir deslocamento e velocidade simultaneamente.

TABELA 2.2 - Relação entre as grandezas físicas e os sensores utilizados em suas medições.

Grandeza física	Sensor
Temperatura	Termômetro/Termopar
Peso	Célula de Carga/Dinamômetro
Velocidade	Tacômetro
Deslocamento	<i>Resolvers</i> (Régua Potenciométrica)
Velocidade/deslocamento	<i>Encoder</i>
Tensão elétrica	Voltímetro
Corrente elétrica	Amperímetro
Resistência elétrica	Ohmímetro
Pressão	Manômetro
Velocidade do ar	Anemômetro

2.2 Robótica industrial

A etimologia da palavra robô vem de uma peça teatral tcheca do início dos anos 20, de autoria de Karel Capek, intitulada “Os Robôs Universais de Rossum”, originando o termo robô (GROOVER, 1989). A palavra tcheca *robota* significa servidão ou trabalho forçado e, quando traduzida para o inglês, transformou-se em *robot*, recebendo a forma robô no português.

A definição oficial de um robô industrial é dada pela Associação das Indústrias de Robótica como um manipulador reprogramável, multifuncional, projetado para mover materiais, peças, ferramentas ou dispositivos especiais em movimentos variáveis programados para a realização de uma variedade de tarefas.

A Associação de Automação e Robótica Australiana (GROOVER, 1989) diz que não existe definição padrão para robô, mas sugere três características essenciais:

- possuir alguma forma de mobilidade;
- ser programável para realizar tarefas diversas;
- operar automaticamente após ser programado.

A anatomia do manipulador robótico ocupa-se da construção física do corpo, braço e punho da máquina. A maioria dos manipuladores robóticos usados hoje em fábricas é montada em uma base que está fixada ao piso. O corpo está ligado à base e o braço ao corpo. Na extremidade do braço, está o punho. O punho consiste em inúmeros componentes que lhe permitem orientação em diversas posições. Movimentos relativos entre os diversos componentes do corpo, braço e punho são proporcionados, geralmente, por juntas. O conjunto formado pela base, braço e punho é por vezes chamado de manipulador. Nos robôs mais novos, o movimento dos braços é baseado no movimento do braço humano, sendo chamados de robôs antropomorfos. Esses robôs possuem seis ou mais graus de liberdade e executam tarefas complexas que necessitam de um difícil posicionamento da ferramenta de trabalho. Um exemplo desses robôs são os utilizados na indústria automobilística (FIG. 2.4), onde a soldagem da carroceria é executada por robôs com seis graus de liberdade.

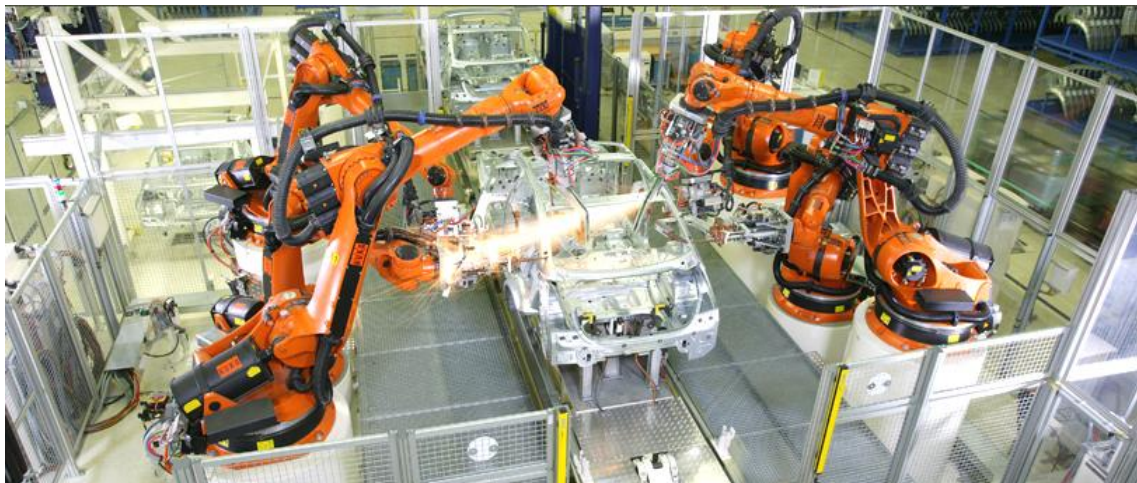


FIGURA 2.4 - Robôs antropomorfos para soldagem de carroceria de automóvel.
FONTE: KUKA, 2012.

2.2.1 Graus de liberdade

Graus de liberdade é um termo genérico utilizado em referência à quantidade mínima de números reais necessários para determinar completamente o estado físico de um dado sistema. Este conceito é geralmente empregado em mecânica clássica e termodinâmica.

Para entender melhor o conceito, é dado o exemplo de um objeto sobre uma mesa (FIG. 2.5). Pela definição, o bloco possui três graus de liberdade, pois pode deslocar-se nas direções cartesianas X e Y e rotacionar no plano XY na direção Z.

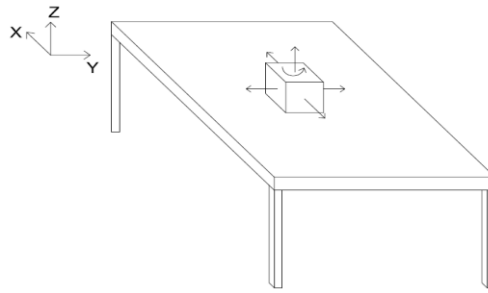


FIGURA 2.5 - Objeto sobre a mesa com três graus de liberdade.

Para descrever a posição de um objeto no espaço, há necessidade de seis parâmetros:

- três parâmetros definem a posição no eixo cartesiano: X, Y e Z;
- três parâmetros definem a orientação do objeto (rolamento (roll) – giro em torno do eixo X, arfagem (pitch) – giro em torno do eixo Y e guinada (yaw) – giro em torno do eixo Z).

Em processos de soldagem *GMAW*, são necessários cinco graus de liberdade para descrever o movimento do TCP da ferramenta, pois é necessário posicionar a tocha nas coordenadas X, Y e Z no espaço e, concomitantemente, orientar o seu ângulo de inclinação. Devido ao fato da tocha de soldagem ser simétrica radialmente e não ser necessário movimentos de rolamento na direção do ângulo de soldagem, sua orientação depende apenas de dois graus de liberdade: puxando ou empurrando o arco elétrico (FIG. 2.6 (a)) e o ângulo em relação ao chanfro (FIG. 2.6 (b)).

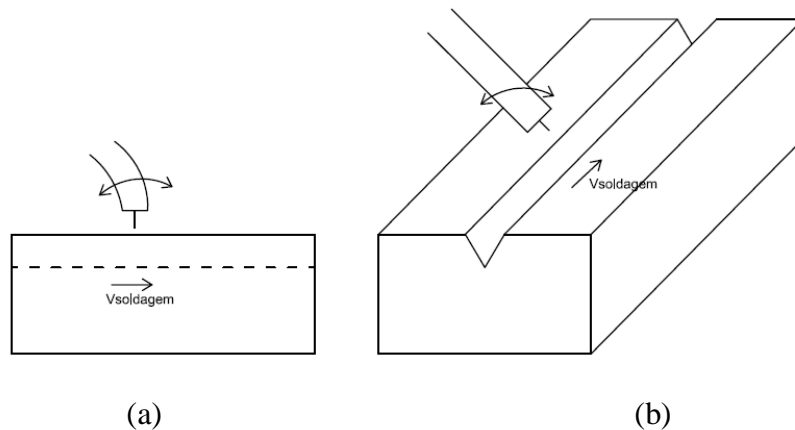


FIGURA 2.6 – Graus de liberdade utilizados para orientação da tocha na soldagem *GMAW*: (a) movimento de puxar e empurrar o arco elétrico e (b) ângulo em relação ao chanfro.

2.2.2 Graus de liberdade em robôs

Em um robô, o número de graus de liberdade corresponde ao número de juntas de movimentos independentes. Um robô de seis graus de liberdade consegue posicionar um objeto ou ferramenta em qualquer posição e orientação no espaço. Se o robô tiver cinco juntas, um dos movimentos da ferramenta será dependente dos outros. Se o robô tiver sete juntas, ele poderá se movimentar sem mudar a posição e orientação da ferramenta, como o braço humano que pode mover o cotovelo mesmo com a mão parada, como pode ser visto pela Figura 2.7.

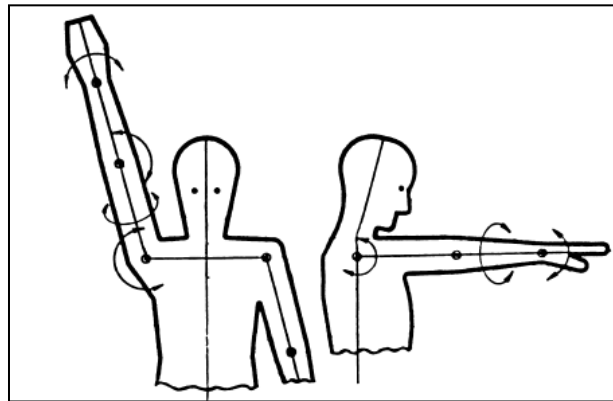


FIGURA 2.7 - Juntas do braço humano totalizando sete graus de liberdade.

2.2.3 Classificação dos robôs

Os robôs industriais podem ser classificados de acordo com o número de juntas, o tipo de controle, o tipo de acionamento e a geometria. É usual classificar os robôs de acordo com o tipo de junta ou, mais exatamente, pelas três juntas mais próximas da base do robô. Também podem ser classificados em relação ao espaço de trabalho, ao grau de rigidez e à extensão de controle sobre o curso do movimento.

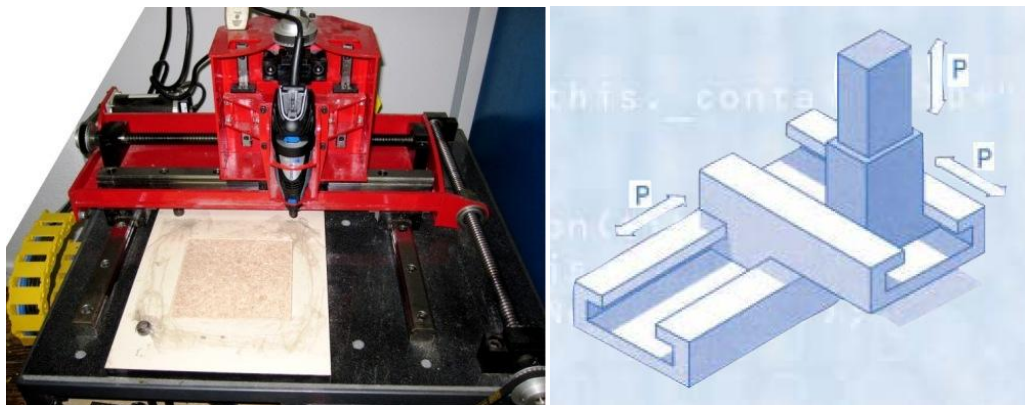
Os diferentes graus de liberdade de um manipulador robótico podem ser encontrados em várias combinações de configurações rotacionais e lineares, dependendo da aplicação. Tais combinações são denominadas geometria do robô (ROMANO, 2002).

Existem seis classes principais de manipuladores, segundo o tipo de junta de rotação (revolução) ou de translação (prismática), o que permite diferentes possibilidades de posicionamento no espaço de trabalho. As seis classes ou geometrias principais de um robô, igualmente chamadas sistemas geométricos coordenados, são: Cartesiana, Cilíndrica, Esférica (ou Polar), de Revolução (ou Articulada), SCARA e Paralela.

Robô de Coordenadas Cartesianas

Os movimentos das juntas prismáticas seguem o plano cartesiano (XYZ). O Robô de Coordenadas Cartesianas, ou Robô Cartesiano, (FIG. 2.8 (a)), pode se mover em linhas retas, em deslocamentos horizontais e verticais. As coordenadas cartesianas especificam um ponto do espaço em função de suas coordenadas X, Y e Z.

Os robôs cartesianos caracterizam-se pela pequena área de trabalho, pelo elevado grau de rigidez mecânica e pela grande exatidão no posicionamento da ferramenta de trabalho. Seu controle é simples em razão do movimento linear dos vínculos, como pode ser visto na Figura 2.8 (b).



(a)

(b)

FIGURA 2.8 - (a) Robô de Coordenadas Cartesianas e (b) movimentos de cada eixo.

FONTE: BRACARENSE *et al.*, 2009.

Robô de Coordenadas Cilíndricas

O Robô de Coordenadas Cilíndricas (FIG. 2.9 (a)) combina movimentos lineares com movimentos rotacionais. Este tipo de robô possui um movimento rotacional na cintura e dois movimentos lineares, que descrevem um cilindro, o que pode ser verificado pela Figura 2.9 (b).

O espaço de trabalho desses robôs é maior que a dos robôs cartesianos, e a rigidez mecânica é ligeiramente inferior. O controle é um pouco mais complicado, quando comparado ao modelo cartesiano, devido à existência de vários momentos de inércia para diferentes pontos no espaço de trabalho e pela rotação da junta da base.

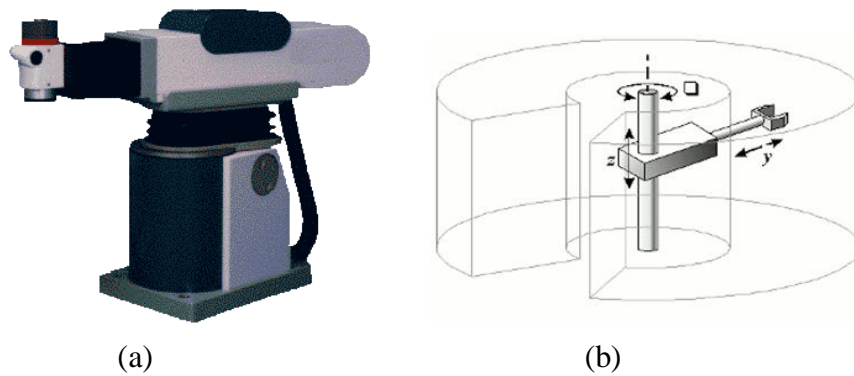


FIGURA 2.9 - (a) Robô de Coordenadas Cilíndricas e (b) seu espaço de trabalho.
 FONTE: BOMFIM, 2009.

Robô de Coordenadas Polares (Esféricas)

Um Robô de Coordenadas Polares (FIG. 2.10 (a)) ou Esféricas possui dois movimentos rotacionais, na cintura e no ombro, e um terceiro que é linear (FIG. 2.10 (b)). Os três eixos descrevem um envelope de trabalho esférico. Seu espaço de trabalho é maior que a dos modelos cilíndricos, e sua rigidez mecânica é menor. O controle é ainda mais complicado devido aos movimentos de rotação.

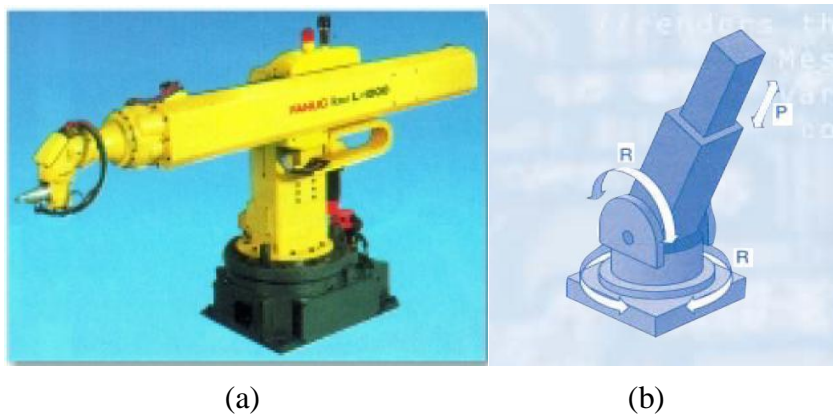


FIGURA 2.10 - (a) Robô de Coordenadas Polares ou Esféricas e (b) movimentos de cada eixo.
 FONTE: ROSÁRIO, 2005.

Robô de Coordenadas de Revolução (Articulado)

As juntas e os movimentos de um Robô de Coordenadas de Revolução ou Articulado se assemelham aos de um braço humano (FIG 2.11 (b)). A Figura 2.11 (a) apresenta o Robô PUMA (*Programmable Universal Machine for Assembly*), um dos projetos mais populares de robôs articulados. Este robô foi projetado, inicialmente, para atender à indústria automobilística.

Sua área de atuação é ampla por permitir alta precisão em posicionamento e orientação da ferramenta de trabalho, apesar de apresentar baixa rigidez mecânica. Seu controle é complicado e difícil em razão das três juntas de revolução e das variações no momento de inércia.

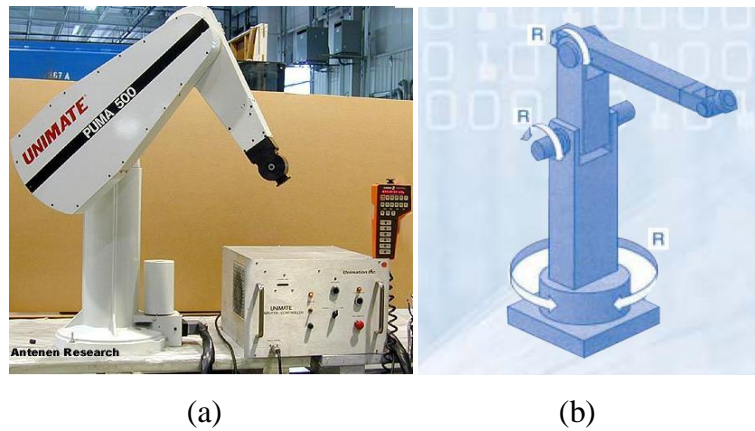


FIGURA 2.11 - (a) Robô PUMA 500 e (b) movimentos de cada junta de um Robô de Coordenadas de Revolução ou Articulado.

FONTE: ROSÁRIO, 2005.

Robô SCARA

O Robô SCARA (*Selective Compliant Articulated Robot for Assembly*) é uma configuração que possui duas juntas de revolução e uma deslizante, como pode ser visto na Figura 2.12 (a). Em termos de configuração física, rigidez mecânica e complexidade de controle, o robô SCARA é similar ao robô cilíndrico, diferenciando-se apenas em questões relacionadas ao espaço de trabalho, o que faz do robô SCARA mais apropriado para tarefas de montagem. Em seu envelope de trabalho, o robô SCARA pode movimentar objetos mantendo sua orientação, o que justifica sua ampla utilização em tarefas de montagem. A Figura 2.12 (b) apresenta os movimentos de cada junta de um Robô SCARA.



FIGURA 2.12 - (a) Robô SCARA modelo TH1050 produzido pela Toshiba e (b) movimentos de cada junta de um Robô SCARA.

FONTE: ROSÁRIO, 2005.

Robô Paralelo

Estes tipos de manipuladores possuem juntas que transformam movimentos de rotação em movimentos de translação, ou usam diretamente juntas prismáticas. Sua principal característica é ter um espaço de trabalho reduzido, porém com uma alta velocidade, o que o torna propício para certas atividades de montagem. Outra característica desses robôs é que eles não possuem cinemática com cadeia aberta como a maioria dos robôs industriais. Na Figura 2.13, é apresentado um exemplo de robô paralelo, onde os quatro atuadores unem a base diretamente ao punho. Devido à alta velocidade e precisão, estes robôs são amplamente empregados em simuladores de veículos, aeronaves e navios.



FIGURA 2.13 - Simulador de dirigibilidade de veículo.

FONTE: TOYOTA, 2012.

Robôs Antropomorfos

Os Robôs Antropomorfos são formados por juntas articuladas e têm seus movimentos baseados no movimento do ser humano, como pode ser verificado pela Figura 2.14 (a). São geralmente aplicados em atividades que necessitam de precisão e complexos movimentos da ferramenta de trabalho. A principal aplicação deste conceito pode ser vista na indústria automobilística, onde a soldagem de carroceria pôde ser automatizada graças ao Robô Antropomorfo. A Figura 2.14 (b) apresenta um robô que possui seis graus de liberdade, o que permite qualquer orientação da ferramenta em qualquer posição do espaço cartesiano.

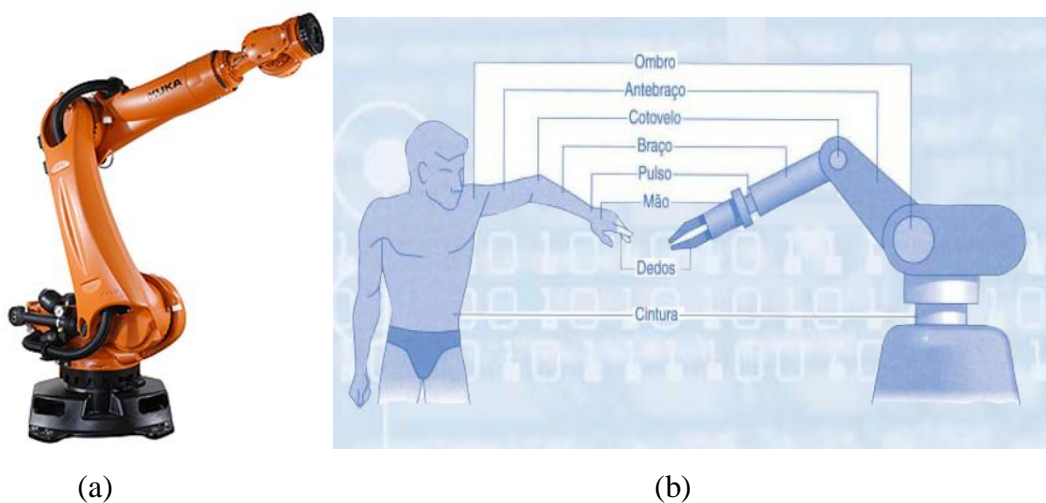


FIGURA 2.14 - (a) Robô Antropomorfo fabricado pela KUKA e (b) as articulações do braço humano e sua relação com as articulações projetadas nas juntas dos Robôs Antropomorfos.

FONTE: ROSÁRIO, 2005.

2.2.4 Braço Robótico

Grande parte dos robôs industriais possui seis graus de liberdade e seus movimentos são baseados no movimento do braço humano. Por isso, são chamados de braços robóticos. Como um braço humano, um manipulador robótico de seis graus de liberdade possui o equivalente a um ombro, um cotovelo e um pulso, no qual tipicamente:

- o ombro é montado em uma base estacionária;
- este tipo de robô tem seis graus de liberdade enquanto que o braço humano tem sete;
- o robô deve mover um efetuador de uma posição à outra. O tipo de efetuador depende da tarefa a ser executada. Alguns tipos comuns são garra, mão antropomórfica ou imã.

Como exemplo, pode-se citar o Robô PUMA 560 (FIG. 2.15) com seis graus de liberdade. Este manipulador robótico possui o equivalente a um ombro, um cotovelo e um pulso, onde o ombro é montado em uma base estacionária.



FIGURA 2.15 - Robô PUMA 560.

FONTE: BOMFIM, 2009.

2.3 Cinemática de Robôs Industriais

Em sua configuração mais básica, um robô industrial é formado por um controlador e um manipulador. O controlador, geralmente do tipo PID, é responsável por melhorar a resposta do sistema de acordo com a sintonia de cada uma de suas ações de comando (proporcional, integral e/ou derivativa) e reduzir erros através da leitura do sinal de realimentação de forma a fazer o manipulador realizar os movimentos programados com boa precisão e resposta rápida. O manipulador é a parte mecânica que realiza o posicionamento mecânico da ferramenta através da movimentação de suas juntas. Os motores de cada junta são acionados por meio de atuadores (*drivers*), que amplificam o sinal de comando oriundo de um computador ou microcontrolador.

Para descrever o posicionamento de um manipulador robótico em um dado momento, podem ser definidos três espaços distintos (CRAIG, 1989):

- Espaço Cartesiano;
- Espaço de Juntas;
- Espaço de Atuadores.

O primeiro modo, Espaço Cartesiano, utiliza o sistema de coordenadas cartesianas para determinar no espaço a posição (x , y e z) e orientação (rolamento, arfagem, guinada) do TCP da ferramenta acoplada ao robô, isto é, em uma determinada posição P serão necessárias seis

coordenadas independentes para descrever o posicionamento. Desta forma, $P(t) = \{x, y, z, r, a, g\}$. O segundo modo, Espaço de Juntas, utiliza o conjunto de variáveis de deslocamento das articulações para definir o posicionamento do manipulador robótico, definido pelo valor dos ângulos de cada junta do robô. Desta forma, $\theta(t) = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\}$. O Espaço de Atuadores é um modo de se definir o posicionamento do robô, considerando os movimentos dos atuadores. Assim, este espaço é dado por $A(t) = \{A_1, A_2, A_3, A_4, A_5, A_6\}$.

Desta forma, ocorre a divisão da cinemática de manipuladores robóticos em cinemática direta e cinemática inversa (FIG. 2.16) sendo a cinemática direta a relação entre o Espaço das Juntas e o Espaço Cartesiano: $\theta(t) \rightarrow P(t)$. Já a cinemática inversa determina a relação entre o Espaço Cartesiano e o Espaço de Juntas: $P(t) \rightarrow \theta(t)$. Estas relações também são válidas quando se trata da cinemática direta e inversa dos atuadores.

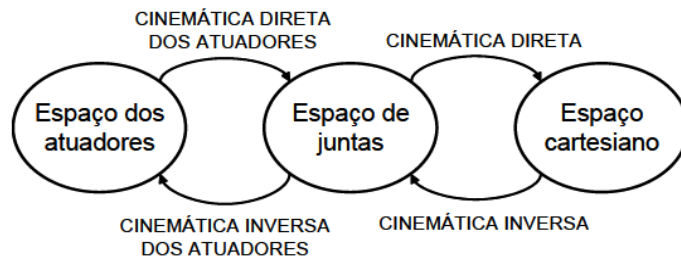


FIGURA 2.16 - Espaços utilizados para definir o posicionamento de um manipulador robótico industrial.

FONTE: LIMA II, 2005.

2.3.1 Descrição matricial para definição de posição e orientação

Para a localização de um ponto no espaço em relação a um sistema de coordenadas $\{A\}$, utiliza-se uma matriz 3×1 , como pode ser verificado na Equação 2.1, denominada vetor de posição (CRAIG, 1989).

$$A_P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (2.1)$$

Para descrever um objeto no espaço, além de definir sua posição, há necessidade de definir também sua orientação. Para isso, define-se um sistema de coordenadas $\{B\}$, preso ao objeto, como pode ser visto na Figura 2.17.

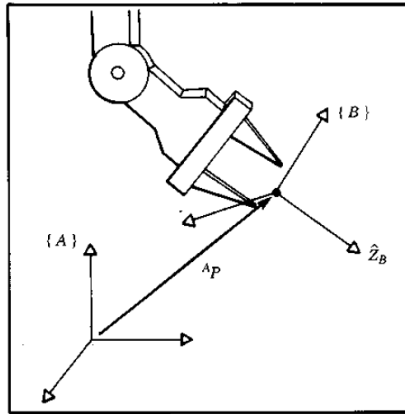


FIGURA 2.17 - Localização de um objeto em posição e orientação.

FONTE: CRAIG, 1989.

Para especificar a orientação do objeto, três vetores são utilizados. Por conveniência, estes vetores são colunas construídas em uma matriz 3x3, onde a orientação é definida a partir dos vetores unitários dos três eixos principais de {B} em relação a {A}, obtendo-se a Equação 2.2.

$${}^A R_B = \begin{bmatrix} A \hat{X}_B & A \hat{Y}_B & A \hat{Z}_B \end{bmatrix} \quad (2.2)$$

Na Figura 2.16, a informação que falta para descrever completamente um sistema de coordenadas {B} em relação a um referencial inercial {A} diz respeito a posição de sua origem (P_{BORG}) em relação a {A}, o que pode ser descrito pela Equação 2.3.

$$\{B\} = \{{}^A R_B, {}^A P_{BORG}\} \quad (2.3)$$

De posse de ${}^A R_B$ e ${}^A P_{BORG}$, é possível transformar a descrição de um ponto em {B} para {A}, como pode ser visto pela Equação 2.4.

$${}^A P = {}^A R_B {}^B P + {}^A P_{BORG} \quad (2.4)$$

Essa equação de transformação pode ser substituída por uma simples multiplicação matricial, através da definição da matriz de transformação homogênea (EQ. 2.5).

$${}^A_B T = \begin{bmatrix} & & & \\ & {}^A_B R & & {}^A P_{BORG} \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Desta forma, para aplicações na área da robótica, a orientação e a posição de um objeto no espaço são representadas por uma matriz 4x4. A Equação 2.6 é um exemplo de uma matriz transformação homogênea de um manipulador robótico de seis graus de liberdade genérico. Os elementos das três primeiras linhas e das três primeiras colunas n, s e a (3 equações de rotação independentes) apresentam três equações de rotação independentes que definem qual será a orientação do TCP do robô em relação à base. Já os elementos da quarta coluna p (3 equações de posição) têm a finalidade de definir qual será a posição do TCP do robô no espaço cartesiano (CRAIG, 1989). A quarta linha (valores triviais) representa uma linha de acoplamento da matriz, permitindo que cálculos matriciais sejam efetuados.

$${}^0 T_6 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

EQUAÇÃO 2.6 - Matriz de transformação homogênea para a definição das coordenadas cartesianas e orientação do TCP de um robô com seis graus de liberdade.

2.4 Hardware de acionamento

2.4.1 Fonte de Alimentação

As fontes de alimentação têm a finalidade de converter a tensão alternada da rede em tensão contínua. É de extrema importância a correta confecção da fonte, cálculo dos transformadores, diodos, banco de capacitores e reguladores de tensão para que sejam atendidas as necessidades de potência dos motores do braço robótico.

A fonte de alimentação tem em sua entrada um sinal que, geralmente, possui um valor eficaz de 127V e 60Hz. Para que a mesma possa atender aos motores, seu sinal de saída deve ser contínuo e ter um valor que atenda a tensão e corrente de partida dos motores. A fonte de alimentação é

constituída basicamente de transformador (trafo), ponte retificadora e banco de capacitores. A função do trafo é elevar ou reduzir a tensão eficaz de saída de acordo com sua relação de espiras. A ponte retificadora tem a finalidade de eliminar o semi-ciclo negativo, tornando o sinal contínuo. Já o banco de capacitores faz com que a tensão de alimentação dos motores tenha baixa oscilação.

A Figura 2.18 apresenta um diagrama em blocos de uma fonte de alimentação. Circuitos eletrônicos que exigem baixa ondulação da tensão de alimentação, como é o caso de circuitos pré-amplificadores de áudio, transmissores de RF e circuitos digitais, deve-se utilizar um circuito regulador de tensão na saída da fonte de alimentação de forma a reduzir o *ripple* da tensão de saída a um valor praticamente nulo.

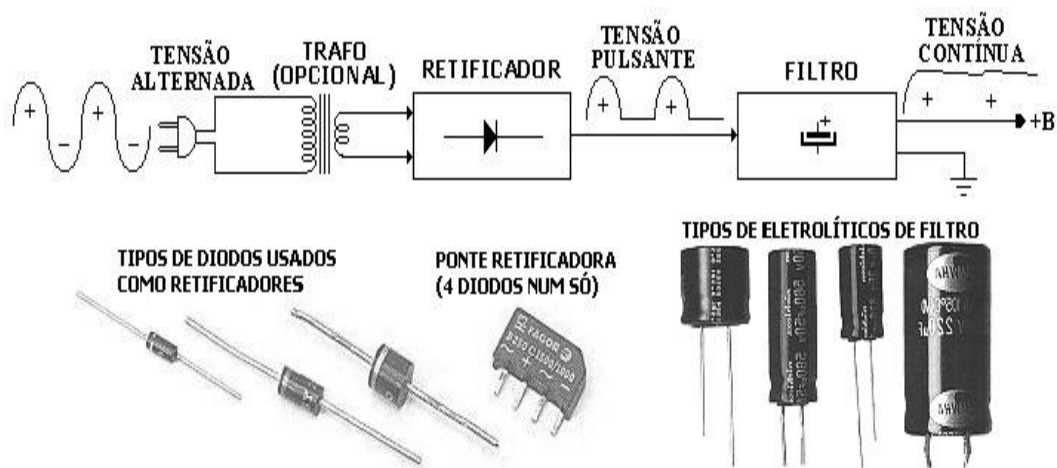


FIGURA 2.18 - Diagrama em blocos de uma fonte de alimentação.

2.4.2 Motores elétricos de corrente contínua (MCC)

A Figura 2.19 apresenta um motor com uma bobina (ou enrolamento de armadura) que gira quando imersa em um campo magnético. O campo magnético (B) é gerado por uma corrente no enrolamento de campo. Quando um condutor (bobina) de comprimento L , circulando uma corrente elétrica i , passa através de um campo magnético B , aparece uma força F sobre o condutor, o que pode ser verificado pela Equação 2.7.

$$F = B \cdot i \cdot L \quad (2.7)$$

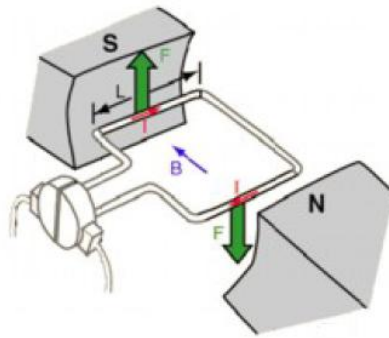


FIGURA 2.19 – Célula básica de um MCC com uma espira.

O MCC é composto basicamente de dois circuitos: circuito de campo e circuito de armadura. O circuito de campo é composto por enrolamentos estáticos que são chamados de estator. O circuito de armadura é composto por enrolamentos conectados ao eixo do motor, denominados rotor. A Figura 2.20 apresenta uma vista em corte de um MCC com seus principais circuitos de operação.

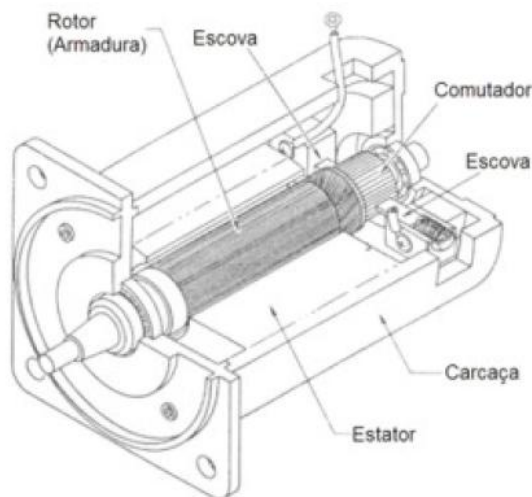


FIGURA 2.20 – Motor elétrico de corrente contínua com seus principais componentes.

O MCC da Figura 2.21 (a) apresenta quatro terminais acessíveis, dois para as bobinas de campo (terminais 3 e 4) e dois para as bobinas de armadura (terminais 1 e 2), como pode ser visto na Figura 2.21 (b). Em alguns motores de baixa potência, as bobinas de campo são substituídas por ímãs permanentes. Neste caso, o motor apresenta apenas dois terminais de acesso (terminais 1 e 2). O princípio de funcionamento elementar de um motor de corrente contínua está baseado na força mecânica que atua sobre um condutor imerso num campo magnético, quando sobre ele circula uma corrente elétrica. Observe que, para este motor funcionar, é necessário inverter o sentido da corrente da armadura a cada 180°, para que o rotor não fique estacionário. Essa

inversão é realizada pelo comutador. A necessidade de comutação é a razão pela qual os enrolamentos de armadura das máquinas de corrente contínua são colocados no rotor. A escova tem a finalidade de realizar o enlace físico entre o rotor e o comutador.

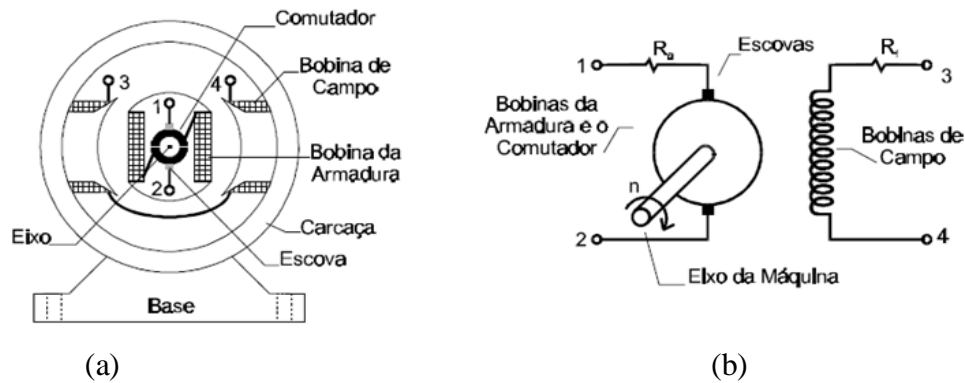


FIGURA 2.21 - (a) Partes internas de um motor de corrente contínua e (b) representação esquemática.

Os MCC eletronicamente controlados, também conhecidos como servomotores, são máquinas construídas para serem aplicadas no controle preciso de posição e velocidade de processos. São motores que possuem resposta mais rápida que os convencionais, pois seus enrolamentos são dimensionados para suportar correntes elevadas e o rotor é projetado com uma baixa inércia, isto é, pequeno diâmetro e grande comprimento. A nomenclatura servomotores deve-se ao fato desses motores serem amplamente utilizados em malha fechada.

Em seu aspecto construtivo, o MCC eletronicamente controlado possui o estator formado por ímãs permanentes e pelas escovas e o rotor é constituído pelas bobinas da armadura e pelo comutador. Esses motores também são conhecidos por motores proporcionais, pois possuem a intensidade de campo do rotor proporcional a potência que o atravessa. Conseqüentemente, a velocidade angular é proporcional à potência aplicada nos terminais da armadura do motor.

2.4.3 Drivers

Nos manipuladores robóticos industriais, os acionadores utilizados são denominados *drivers*. Sua função básica é controlar a energia que alimenta os motores, através do sinal de comando oriundo do computador. O *driver* (FIG. 2.22) pode ser definido como uma chave de estado sólido (transistor), no qual sua modulação é controlada por um sinal de baixa potência do *software* que gerou a cinemática (KENJO, 1991). Basicamente, existem dois tipos de modulações aplicadas a este tipo de controle: modulação por largura de pulsos (*PWM*) e

modulação por pulsos. Assim como os motores, os *drivers* também podem ter a denominação servo (*servodrivers*), quando são aplicados em sistemas de controle em malha fechada.



FIGURA 2.22 - Servodriver modelo G320 fabricado pela Geckodrive.

FONTE: GECKODRIVE, 2012.

A modulação por largura de pulso (*PWM*) permite que a tensão eficaz de saída seja controlada através da modulação do ciclo de trabalho de uma fonte de alimentação. Com este tipo de controlador, evita-se o desperdício de potência que era verificado nos controladores que utilizavam reostatos em série com o motor para ajuste de velocidade (VORPÉRIAN, 1990).

Na Figura 2.23, é apresentado o sinal de tensão de entrada de um motor proporcional com ciclo de trabalho de 30%. Se este motor possui tensão nominal de 10V e rotação máxima de 1000rpm e sua fonte de alimentação for modulada com o ciclo supramencionado, o motor receberá na entrada da sua armadura um sinal chaveado, onde a fonte de alimentação ficará 30% do tempo ligada e 70% do tempo desligada, como pode ser verificado pela Figura 2.23 (a). Desta forma, o motor entenderá que está sendo alimentado com 3V e sua rotação será de 300rpm (FIG. 2.23 (b)). Devido a efeitos inerciais do rotor, o motor se comporta como se recebesse uma forma de onda contínua.

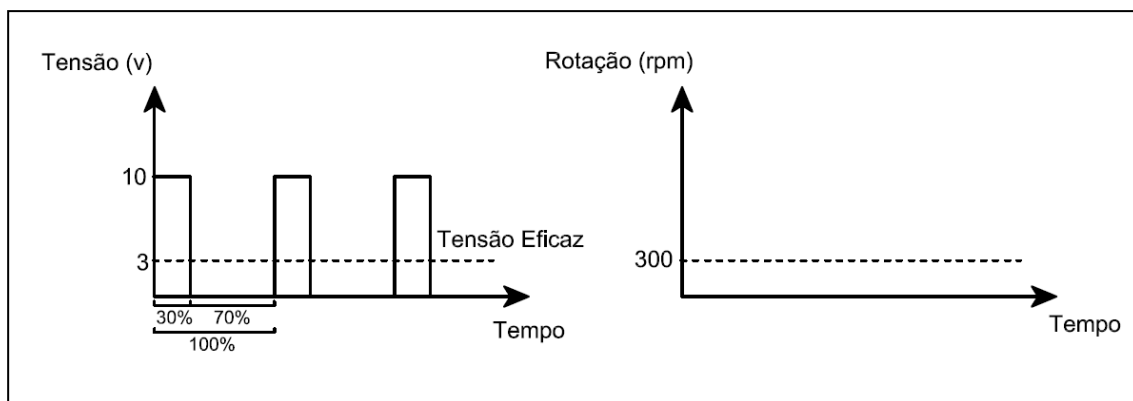


FIGURA 2.23 - (a) Sinal de controle de um motor com ciclo de trabalho de 30% e (b) rotação desenvolvida pelo motor.

Na modulação por pulsos, cada movimento do motor está condicionado à quantidade de pulsos que é necessário para que o eixo faça o deslocamento pré-determinado. De acordo com a quantidade e a taxa (pulsos/segundo) com que os pulsos são gerados, pode-se ter um controle preciso de posicionamento, velocidade e aceleração. A modulação por pulsos é amplamente utilizada em sistemas que utilizam motores de passo para acionamento.

2.4.4 Sensores

Nos manipuladores robóticos, o acionamento das juntas é feito a partir de motores que exigem um controle em malha fechada. Normalmente, estes motores têm sua velocidade de rotação e posicionamento mensurados através de sensores ópticos (*encoders*). Nestas máquinas, até a década de 90, o controle de realimentação de velocidade era feito por tacômetros e o de posição por *resolvers*, o que tornava o tratamento de sinais uma tarefa complicada, pelo fato de se ter dois sinais distintos de realimentação. Para simplificar a malha de controle, foram criados os *encoders*. Os *encoders* são sensores acoplados ao eixo do motor que realimentam pulsos de acordo com o giro de eixo. Desta forma, os sinais que são emitidos pelo *encoder* podem ser comparados diretamente pelo sinal de controle, modulado por pulsos, tornando o controle do sistema mais simples (ROSÁRIO, 2005).

Os *encoders* são o padrão das máquinas mais novas e oferecem um sistema de realimentação robusto (óptico), digital, discreto e de baixo custo, como pode ser visto na Figura 2.24 (a).

O *encoder* é composto por um detector óptico duplo com o espaçamento entre eles menor que uma franja do disco graduado, de forma a permitir a detecção não só da frequência, mas também da direção de rotação (FIG. 2.24 (b)).

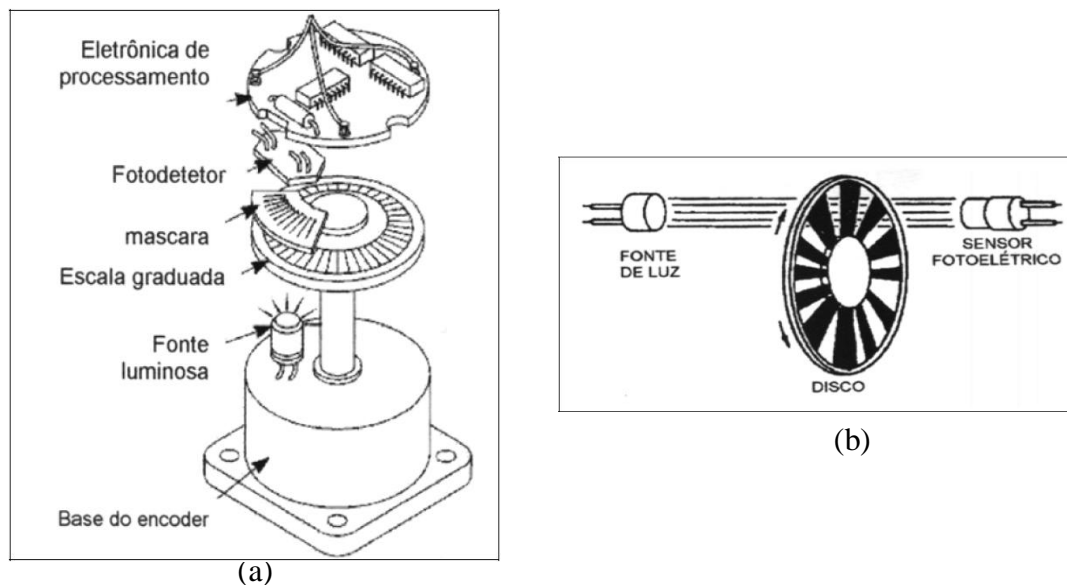


FIGURA 2.24 - (a) Vista explodida de um *encoder* e (b) detector óptico e disco graduado.

FONTE: BOMFIM, 2009.

Na construção da cinemática dos manipuladores robóticos, a resolução do *encoder* é a peça chave para saber os limites de velocidade, aceleração e precisão que um sistema pode ter. À medida que se aumenta a resolução de um *encoder*, a sua precisão é melhor. Em detrimento do aumento da precisão, serão necessários mais pulsos para manter os mesmos padrões de velocidade e aceleração. Por exemplo, se um *encoder* tiver uma resolução de 512 pulsos por volta e estiver realizando o controle de realimentação de uma fresa CNC, cada volta do motor representará um deslocamento de 10mm. Para que esta ferramenta desloque 100mm em x, com velocidade de 10mm/s, terão que ser gerados 5.120 pulsos a uma taxa de 512 pulsos/segundo. Verifica-se neste sistema que a precisão está inadequada e a velocidade está baixa. Consequentemente, o *encoder* deve ser trocado por outro com resolução de 8.192 pulsos/volta, melhorando a precisão em 1600%. Para que a velocidade seja de 50mm/s e deslocamento de 100mm, serão necessários 81.920 pulsos a uma taxa de 40.980 pulsos/segundo (40,98kHz). Desta forma, deve-se tomar cuidado para não extrapolar os limites do sistema, isto é, operar em frequências acima das recomendadas pelos fabricantes de *drivers*, portas paralelas e *softwares* utilizados no controle de máquinas CNC ou manipuladores robóticos.

Na Figura 2.25 (a), há uma vista explodida de um *encoder* fabricado pela USDigital, amplamente utilizado em sistemas de controle. Em sua constituição básica, este sensor é formado pelo módulo de transmissão (FIG. 2.25 (b)) e pelo disco de *encoder* (FIG. 2.25 (c)). O módulo de transmissão tem em sua carcaça a eletrônica de processamento, fotodetector e a fonte luminosa.

Já o disco possui a escala graduada. Quando o *encoder* é acionado, o módulo de transmissão passa a emitir um sinal contínuo, através de um feixe de laser, que é recebido por um fototransistor. À medida que o eixo do motor gira, o disco com a escala graduada também gira, fazendo com que a luz do laser seja cortada toda vez que uma escala passa em sua frente. Assim, são gerados pulsos que, após serem processados, irão realimentar a malha de controle.

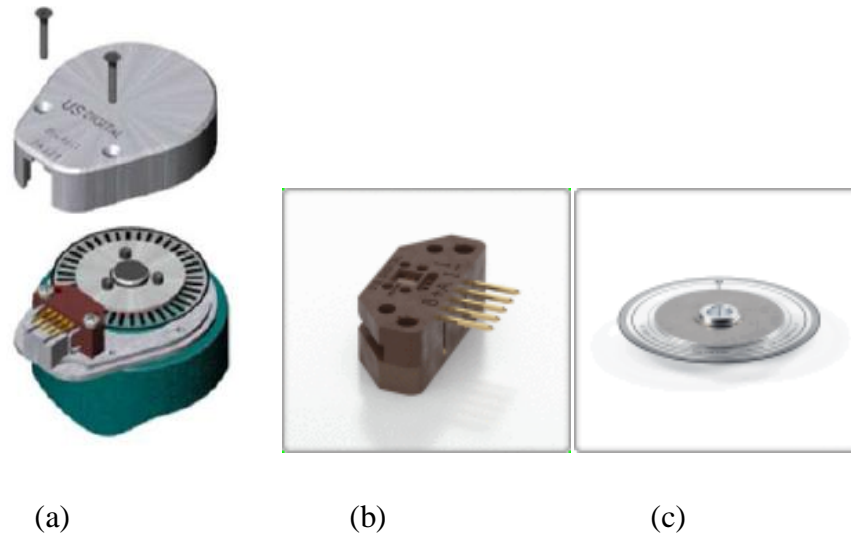


FIGURA 2.25 - (a) *Encoder* fabricado pela USDigital, (b) módulo de transmissão do *encoder* óptico e (c) disco de *encoder*.

FONTE: USDIGITAL, 2012.

2.4.5 Portas de interface paralela e *USB*

A porta paralela é uma interface de comunicação entre dois computadores ou um computador e um periférico. Quando a IBM criou seu primeiro computador, a ideia era conectar essa porta a uma impressora, mas, atualmente, são vários os periféricos que podem utilizar desta conexão para enviar e receber dados para o computador.

A partir do sistema operacional Windows 95, tornou-se possível efetuar comunicação entre dois computadores através da porta paralela. Esta rede é muito simples de ser implementada, bastando apenas a utilização de um conector DB-25 (FIG. 2.26 (a)). É, no entanto, necessária uma configuração específica nos cabos para que a rede possa funcionar corretamente.

Na comunicação em paralelo, grupos de bits são transferidos simultaneamente, através de diversas linhas condutoras dos sinais. Desta forma, como vários bits são transmitidos simultaneamente a cada ciclo, a taxa de transferência de dados é alta.

Atualmente, as portas paralelas vêm sendo substituídas por interfaces onde os dados são transmitidos de forma serial, denominadas portas *USB*. Na comunicação serial, dados são transmitidos através de uma única linha condutora (barramento serial). A primeira vista, uma porta paralela seria mais rápida que uma *USB*, mas o que ocorre é o oposto, porque foram criados protocolos de comunicação serial de dados com velocidades que podem chegar a 4.8 GBps (*USB 3.0*), contra 3 MBps da porta paralela (CNC4, 2012).

A Figura 2.26 (b) apresenta a interface *USB*. Esta interface foi projetada de maneira que possam ser ligados vários periféricos pelo mesmo canal. Assim, mediante uma topologia em árvore, é possível ligar até 127 dispositivos a uma única porta do computador. Esta topologia pode ser implementada com a utilização de *Hubs* e *Switches* (SILVA, 2003).

Uma das desvantagens da porta *USB* é que o preço de equipamentos com esta tecnologia é maior que periféricos com porta paralela, devido a sua maior complexidade de construção. Outro fator que torna a porta *USB* menos atrativa é o fato do computador que está gerenciando os periféricos necessitar de um *firmware* (popularmente conhecido como *driver*) específico para o sistema operacional que está sendo utilizado. Este *firmware* é um *software* com a função de gerenciar o tráfego de dados através do protocolo de comunicação.

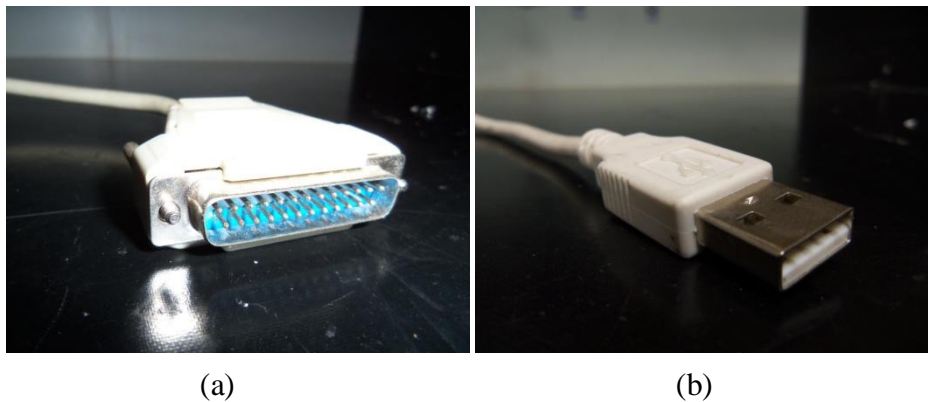


FIGURA 2.26 - (a) Porta paralela e (b) porta *USB*.

2.4.6 Gabinetes de controle de manipuladores robóticos industriais

Um gabinete de controle é formado basicamente de quatro partes: interface entre *drivers* e computador, *drivers*, controlador e fonte de alimentação (BOMFIM, 2009). A função da interface é fazer com que o gabinete se comunique com um controlador dedicado. O *driver* recebe o sinal enviado pelo computador e transforma em pulsos para acionamento dos motores. A fonte de alimentação deve gerar uma tensão com potencial igual à tensão nominal de alimentação dos motores, atendendo as necessidades de potência. O controlador, geralmente, está

embutido no *driver* e seu ajuste pode ser feito através de potenciômetros que controlam as ações de comando proporcional, integral e derivativa (controladores PID).

A Figura 2.27 apresenta uma visão frontal de um gabinete de controle fabricado pela ABB para controlar o manipulador robótico modelo IRB 2000. Um dos problemas encontrados para a realização de manutenção e incorporação de mudanças no sistema, como a adaptação de novos sensores, por exemplo, é que os diversos fabricantes como KUKA, Fanuc, ABB, COMAU, Fuji Robotics e Motoman não possuem intercambialidade entre si, ou seja, um gabinete de controle não pode ser utilizado entre fabricantes distintos e os fabricantes não seguem uma padronização no projeto e montagem de seus sistemas. Desta forma, para incorporar mudanças no processo, os usuários têm que adquirir todo o sistema do mesmo fabricante do robô, além de contratar técnicos que saibam como programar os protocolos e interfaces proprietárias, o que encarece muito a implementação de mudanças ao sistema.



FIGURA 2.27 - Gabinete de controle do manipulador robótico modelo IRB 2000 fabricado pela ABB.

FONTE: GLEIZER, 2010.

Assessórios para manipuladores robóticos, como placas de I/O e de comunicação não custam muito mais do que placas similares direcionadas ao mercado de computadores, mesmo considerando-se os efeitos das diferenças nas escalas de produção. Uma das razões para estes altos custos é a falta de uma arquitetura aberta para os controladores de manipuladores robóticos. Cada fabricante possui seus protocolos e interfaces próprios, forçando os usuários a adquirir todo o sistema do mesmo fabricante. Desta forma, e seguindo uma tendência mundial de padronização

em automação e sistemas de controle, foi desenvolvido no LRSS/UFMG um gabinete de controle com interface paralela para ser aplicado em manipuladores robóticos com até seis graus de liberdade (BOMFIM *et al.*, 2012). Este gabinete de controle, apresentado na Figura 2.28, é capaz de controlar qualquer manipulador desde que o mesmo tenha até seis juntas (seis graus de liberdade e um motor conectado a cada grau) e a fonte de alimentação atenda aos requisitos de potência e tensão dos motores.



FIGURA 2.28 - Gabinete de controle com interface paralela fabricado no LRSS.

FONTE: BOMFIM *et al.*, 2012.

2.5 Máquinas com Comando Numérico Computadorizado (CNC)

Com a evolução contínua de sistemas microprocessados, ocorre cada vez mais o aumento da taxa de processamento, juntamente com uma redução de tamanho e preço dos computadores. Devido a sua capacidade de repetibilidade, elevado processamento e a possibilidade de criação de interfaces cada vez mais amigáveis com o operador, o computador é amplamente difundido em todos os setores da sociedade moderna.

No setor industrial, esta evolução contínua vem acompanhada da substituição das máquinas de operação manual por máquinas operadas por sistemas automáticos ou semi-automáticos. Neste contexto, passaram a ser desenvolvidas, a partir da década de 70, as máquinas CNC, Comando Numérico Computadorizado. Em um primeiro momento, este novo conceito foi aplicado principalmente às máquinas destinadas a processos de usinagem, onde havia o controle computadorizado da velocidade e profundidade com que material era retirado da peça a ser usinada. Atualmente, as máquinas CNC são aplicadas em processos de torneamento, fresamento, corte a plasma, dentre outros processos.

2.5.1 Componentes de uma máquina CNC

Os componentes principais de um sistema CNC são apresentados na Figura 2.29. Esses componentes são basicamente: programa do projeto assistido por computador, meio de transferência de dados, controlador da máquina, placas controladoras (atuadores) de potência e uma máquina/ferramenta. No programa do projeto assistido por computador, o programador utiliza o programa *CAD/CAM* para gerar um arquivo de saída chamado “*G-code*”. O programa descreve as etapas da máquina exigidas para fazer a peça desejada. O meio de transferência de dados é uma forma de interligar a saída do programa *CAD/CAM* ao controlador da máquina. Este meio de transferência pode ser um *pendrive*, uma porta paralela ou *USB*, ou até mesmo uma rede de internet ou intranet. O controlador da máquina lê e interpreta o programa da peça para controlar a ferramenta que cortará a mesma. Desta forma, o *software* controlador da máquina (Mach 3, por exemplo) funcionará em um computador, executando a função de interpretador, recebendo os comandos do Código G e emitindo os sinais para cada atuador, para o controle de posicionamento, velocidade e aceleração de cada eixo cartesiano. Os sinais emitidos pelo computador são de baixa potência, geralmente de 5V e alguns mW. Consequentemente, há a necessidade da utilização de placas controladoras de potência. A função básica destas placas é modular uma saída de potência elevada, necessária para alimentação dos motores, através de um sinal de baixa potência oriundo do computador. Por último, existe a máquina/ferramenta que é fixada ao TCP da máquina CNC.

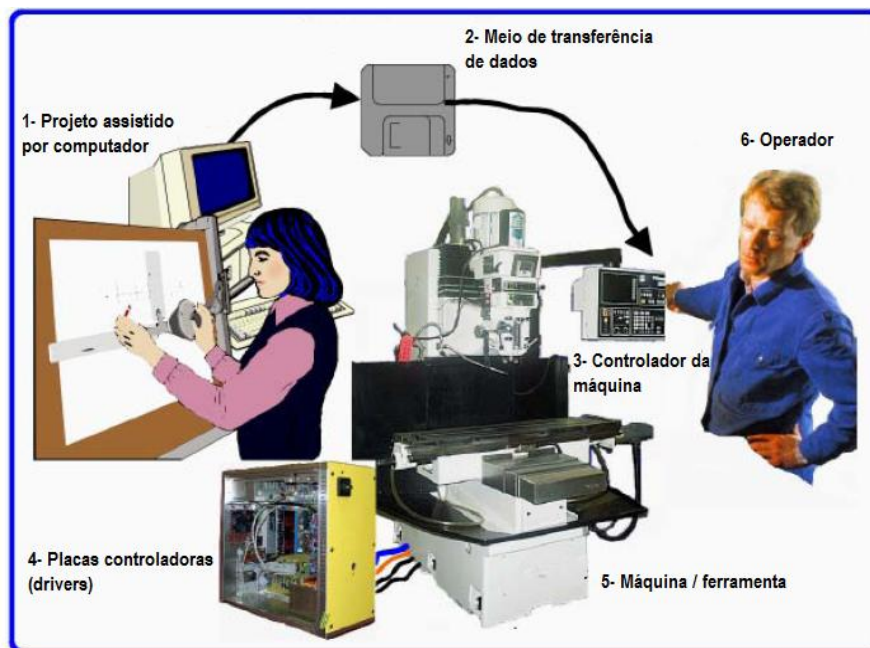


FIGURA 2.29 - Componentes básicos de uma máquina CNC.

ADAPTADO DE: MACHSUPPORT, 2012.

2.6 Retrofitting

Os primeiros manipuladores robóticos industriais começaram a ser produzidos em larga escala na década de 70, para aplicações relacionadas, principalmente, com a produção na indústria automobilística (CRAIG, 1989). Desde então, o número de robôs que estão sendo “encostados” no chão de fábrica cresce a cada dia, devido ao fato da escassez de mão-de-obra qualificada para a execução de reparos, além de, geralmente, o sistema de controle do robô ter uma tecnologia obsoleta. Desta forma, atualizações do sistema e modificações na estrutura já existente tornam-se não compensatórias, pois o custo é muito elevado.

Uma solução que vem sendo aplicada desde os anos 90 nos Estados Unidos, principalmente a equipamentos CNC, é o processo conhecido como *retrofitting*. A etimologia deste novo conceito vem das palavras “revisar” e “atualizar”. O processo de *retrofitting* divide-se basicamente em duas etapas, onde na primeira etapa a mecânica do equipamento CNC é revisada, sendo avaliado se é ou não compensatório colocar o equipamento novamente em operação. Na segunda etapa ocorre a atualização da máquina CNC a novas tecnologias, com o reaproveitamento de peças antigas de forma a reduzir o custo do processo.

Dependendo do nível de modernização que se queira executar, pode-se optar pela simples troca do CNC ou também dos seus servoacionamentos. A modernização pode também envolver trocas ou aplicações de conjuntos mecânicos como fusos, redutores, sistemas de lubrificação e outros. A princípio, não é necessário efetuar uma reforma mecânica do equipamento, desde que a estrutura mecânica da máquina esteja em condições adequadas (BRACARENSE *et al.*, 2003).

Atualmente, é prática comum a execução de *retrofitting* em máquinas relacionadas com processos de usinagem (máquinas de torno-fresamento) e com operações de *pick and place*. Estas máquinas são remanufaturadas, geralmente, por se tratarem de equipamentos antigos, dotados de sistema de controle completamente obsoletos, no qual o mau funcionamento, a frequente intervenção da manutenção e a inexistência de peças de reposição, oneram significativamente a produção (LIMA II *et al.*, 2007). Basicamente, existem três fatores que motivam a execução do *retrofitting*:

1. Sistemas de controle operando em boas condições, mas com necessidade de realizar adaptações no sistema e/ou adequações a novas tecnologias;
2. Equipamento obsoleto, com mau funcionamento e inexistência de peças de reposição e falta de pessoal técnico especializado na execução de reparos;

3. Máquinas de controle manual que são automatizadas através da implantação de sistemas de controle automatizados.

Neste contexto, ocorre, principalmente, o *retrofitting* de máquinas que possuem até três graus de liberdade, devido a cinemática desses equipamentos ser de fácil programação e existirem *softwares* de geração de cinemática amplamente difundidos no mercado, para aplicações onde o controle é feito por CNC ou CLP. O *retrofitting* de manipuladores robóticos com mais de quatro graus de liberdade não é prática comum, pois a programação da cinemática não é uma tarefa simples. Em robôs com três graus de liberdade, geralmente, o movimento de cada junta é vinculado a um eixo cartesiano. Já em situações com mais de três graus de liberdade o movimento em cada coordenada cartesiana depende do movimento de várias juntas. Assim, é necessário descrever o movimento do TCP do manipulador em forma de matrizes, pois, nestas situações, ocorre o movimento cartesiano e, concomitante, o movimento de orientação da ferramenta de trabalho.

Nas três próximas subseções, serão dados exemplos de cada um dos três fatores que motivam a execução do *retrofitting*.

2.6.1 Realização de adequações a novas tecnologias em manipuladores robóticos industriais

Gleizer (2010) desenvolveu o *retrofitting* de um robô industrial modelo IRB2000, fabricado pela ABB. Este *retrofitting* teve a finalidade de atualização de *hardware* e *software* para que o manipulador robótico pudesse ser controlado a partir de uma linguagem de programação não proprietária. Para tanto, ocorreu a construção de um novo gabinete de controle, como pode ser visto na Figura 2.30 (a).

Aplicando a filosofia do *retrofitting*, observou-se que era viável o reaproveitamento da estrutura mecânica do robô, juntamente com motores, freios e sensores. O gabinete de controle antigo (FIG. 2.30 (b)) teve que ser trocado para permitir que os atuadores fossem acionados por servoconversores comerciais. Estes servoconversores são conectados a um computador através de uma rede *CAN-Bus* de dados. O algoritmo de controle é implementado através de uma plataforma Simulink/MatLab, a partir do desenvolvimento de uma biblioteca de comunicação e uma interface homem-máquina. Isso permite que o usuário realize tarefas simples com o robô como, por exemplo, o controle independente das juntas, o controle de posição e calibração de parâmetros do sistema. (GLEIZER, 2010).



(a)

(b)

FIGURA 2.30 - (a) Novo painel de controle utilizado na atualização de *hardware* de um robô IRB 2000 e (b) painel de controle original.

FONTE: GLEIZER, 2010.

A Figura 2.31 apresenta a arquitetura de controle utilizada na realização da atualização de *hardware* e *software* do robô IRB2000.

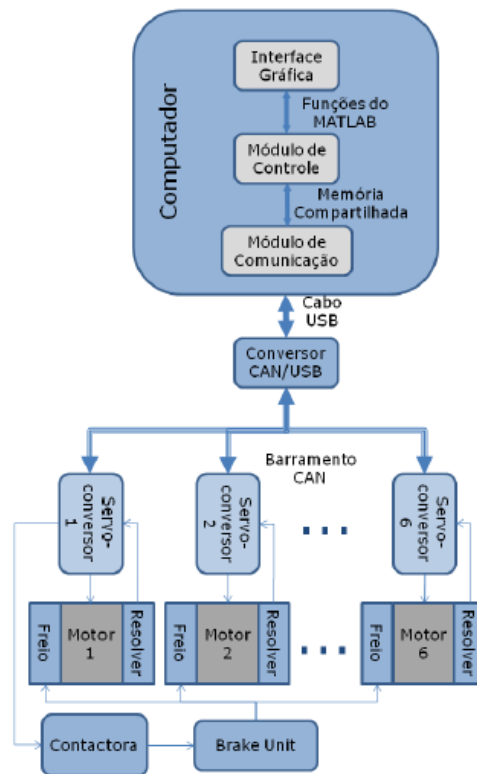


FIGURA 2.31 - Diagrama em blocos da arquitetura de controle desenvolvida.

FONTE: GLEIZER, 2010.

Os resultados atingidos mostraram que o *retrofitting* é uma alternativa viável, pois a atualização tecnológica pôde ser realizada com a reutilização da estrutura mecânica, dos atuadores e dos sensores de posição. Desta forma, foi desenvolvida uma arquitetura de controle aberta, onde a programação da trajetória do robô é realizada através de uma linguagem não proprietária.

2.6.2 Revitalização de máquinas obsoletas, com mau funcionamento e inexistência de peças de reposição

Em trabalhos desenvolvidos por Lima II *et al.* (2007), foi apresentada a proposta de execução do *retrofitting* de uma máquina de separação de fardos de jornal, denominada IDAB *Marathoner Stacker*, ou simplesmente *Stacker*, conforme apresentado na Figura 2.32. A atualização do equipamento foi proposta por se tratar de uma máquina antiga com sistema de controle obsoleto, gerando uma manutenção onerosa devido à inexistência de peças de reposição.



FIGURA 2.32 - Máquina IDAB *Marathoner Stacker*.

FONTE: LIMA II *et al.*, 2007.

A máquina *Stacker* era constituída de um sistema de controle e uma placa eletrônica dedicada com arquitetura fechada. Para o posicionamento do dispositivo que recolhe os fardos dos jornais, era utilizado um servomotor dedicado para esta operação, o qual não podia sofrer qualquer tipo de intervenção, não permitindo flexibilidade para o posicionamento do dispositivo. A IHM era constituída de um protocolo proprietário para comunicação com a placa de controle central. Desta forma, modificações na programação da máquina teriam que ser feitas pelo fabricante.

A arquitetura proposta para a realização do *retrofitting* na máquina *Stacker* foi a utilização de um CLP, uma IHM e um servomotor conectado ao CLP, para realizar o posicionamento do dispositivo que recolhe o jornal. O CLP foi utilizado como controlador do sistema, ou seja,

através da linguagem *ladder* foram estabelecidos comandos de acordo com as necessidades do cliente, o que não era possível realizar com o sistema de controle antigo. O controlador tem a finalidade de modular o funcionamento do motor a partir dos comandos do CLP. Para que o sistema possibilitasse uma maior flexibilidade, foi inserida uma IHM, para permitir ao operador realizar modificações na quantidade de jornais que são separados em cada fardo (LIMA II *et al.*, 2007). A Figura 2.33 apresenta o diagrama da arquitetura utilizada.

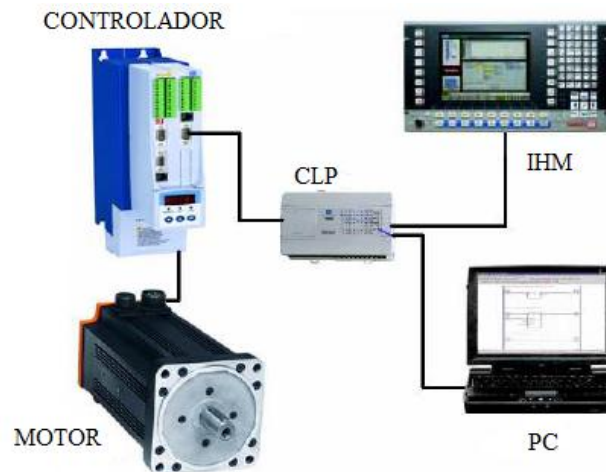


FIGURA 2.33 - Diagrama da arquitetura utilizada no *retrofitting*.

FONTE: LIMA II *et al.*, 2007.

Nesse processo de remanufatura, foi verificado que o método proposto de *retrofitting* constitui uma ferramenta eficaz no desenvolvimento de estratégias de controle para equipamentos que mantém estrutura eletromecânica em boas condições, mas estão com o controle obsoleto (LIMA II *et al.*, 2007). Outra característica observada é que a utilização de uma linguagem não proprietária facilita a realização de modificações na programação do equipamento, originando uma independência do cliente perante o fabricante.

2.6.3 Automatização de máquinas de controle manual

Em trabalhos desenvolvidos por Ferraresi (2011), um torno de controle manual foi automatizado através da implantação de um sistema de controle CNC. O trabalho descreve a construção de um CNC didático, de baixo custo, a partir da modernização de um torno mecânico convencional e a utilização de um microcomputador como unidade de comando numérico da máquina. Os comandos são enviados pelo computador através de uma interface paralela aos motores de passo (FIG. 2.34 (a)). Estes, por sua vez, realizam movimento de rotação que permitem o deslocamento linear dos carros do torno. O *Software* adotado foi o Mach 3, comercializado pela ArtSoft. Sua

escolha baseou-se em algumas vantagens como importação direta de arquivos CAD, geração de códigos G, facilidade de programação através do bloco de notas e a possibilidade de aproveitar os computadores tipo desktop comerciais (FERRARESI, 2011).

Na Figura 2.34 (b), é apresentado o torno sendo controlado pelo *Software* fabricado pela ArtSoft. Seguindo o princípio do *retrofitting*, a estrutura mecânica do torno foi completamente revisada e, após a troca de poucos componentes (substituição dos acionamentos mecânicos do torno e adequação de três castanhas), ficou atestado que seria viável o seu reaproveitamento. Em uma segunda etapa, ocorreu a atualização do torno convencional a novas tecnologias com o reaproveitamento de peças antigas, de forma a reduzir o custo do processo. Por se tratar de um equipamento de controle manual, poucos componentes eletrônicos foram reaproveitados sendo que toda a estrutura de *hardware* foi refeita. Os componentes que foram reaproveitados são os que fazem parte do acionamento do motor principal. O motor, que é acionado em partida direta, utiliza somente um enrolamento (velocidade mais alta). O sinal de comando para a partida é transmitido através da porta paralela.



FIGURA 2.34 - (a) Gabinete de controle utilizado no *retrofitting* e (b) torno mecânico convencional controlado por CNC.

FONTE: FERRARI, 2011.

Este trabalho de *retrofitting* de um torno convencional (manual) alcançou de forma satisfatória seus objetivos, pois com um baixo custo (R\$ 6.000,00), foi possível atualizar as configurações de

forma que o torno apresentasse características equivalentes as de um torno com valor de mercado de R\$ 120.000,00.

Analisando os três processos de *retrofitting*, pode-se chegar à conclusão que a metodologia proposta é realizar adaptações a novas tecnologias com o reaproveitamento de componentes antigos, tornando o processo de baixo custo. Os esforços dos pesquisadores são criar uma arquitetura de controle aberta, com linguagens de programação não proprietárias, de forma a tornar o gabinete de controle intercambiável com qualquer equipamento e fazer com que modificações e implementações ao sistema sejam independentes de fabricantes. Neste contexto, o *retrofitting* passa a ser uma solução para a reciclagem de máquinas inservíveis. Cabe ressaltar que apesar de um *retrofitting* custar em torno de 5% de um equipamento novo (FERRARESI, 2011), é necessário avaliar se a precisão da máquina “retrofitada” será satisfatória.

O aquecimento do mercado mundial de máquinas-ferramenta provocou aumento de vendas de serviços de *retrofitting*, especialmente para máquinas de grande porte e/ou especiais, pois os prazos de entrega de uma nova chegam a superar os 12 meses. No caso de máquinas de menor porte e *standard*, a opção pelo *retrofitting* deve ser bem analisada porque a relação de custo/benefício pode não justificar o investimento. Porém, às vezes, o usuário opta pelo *retrofitting* devido à escassez de máquinas novas para entrega dentro do prazo necessário ou alguma peculiaridade específica do equipamento (BOMFIM, 2009).

Em épocas em que custo e prazo são itens cruciais de competitividade, o *retrofitting* passa a ser, no mínimo, uma alternativa a ser considerada quando das decisões relativas a novos investimentos para o aumento da capacidade produtiva e só uma análise detalhada poderá revelar o que pode ser mais vantajoso, comprar novo ou “retrofitar”.

2.7 Processo de soldagem a arco com arame tubular

A soldagem a arco com arame tubular (*Flux-Cored Arc Welding – FCAW*) é um processo de soldagem que emprega eletrodo sólido contínuo sob proteção gasosa. A coalescência das peças metálicas é produzida pelo aquecimento destas com um arco elétrico estabelecido entre um eletrodo tubular, contínuo, consumível e a peça de trabalho (BRACARENSE, MARQUES & MODENESI, 2007). A proteção do arco e da região da solda contra contaminação pela atmosfera é feita por um gás ou mistura de gases, que podem ser inertes ou ativos, como pode ser visto na Figura 2.35.

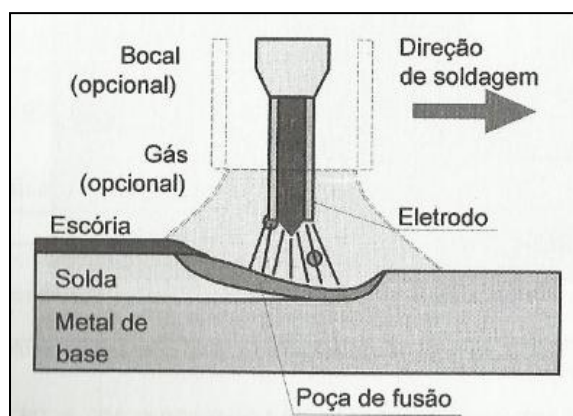


FIGURA 2.35 - Soldagem *FCAW* (esquemática).

FONTE: BRACARENSE, MARQUES & MODENESI, 2007.

O calor gerado pelo arco elétrico é usado para fundir as peças a serem soldadas e o arame eletrodo, que é transferido para a junta e constitui o metal de adição. O modo de transferência do metal fundido na ponta do eletrodo para a poça de fusão é vital na soldagem *FCAW*, pois afeta muitas das características do processo, como a estabilidade do arco, a quantidade de gases absorvida pelo metal fundido, a aplicabilidade do processo em determinadas posições de soldagem e as características do cordão de solda, como o nível de respingos gerados.

O equipamento básico de uma soldagem *FCAW* convencional consiste de uma fonte de energia regulável, de saída tipo tensão constante, tocha de soldagem, fonte de gás e um alimentador de arame regulável de velocidade constante, conforme esquema mostrado na Figura 2.36.

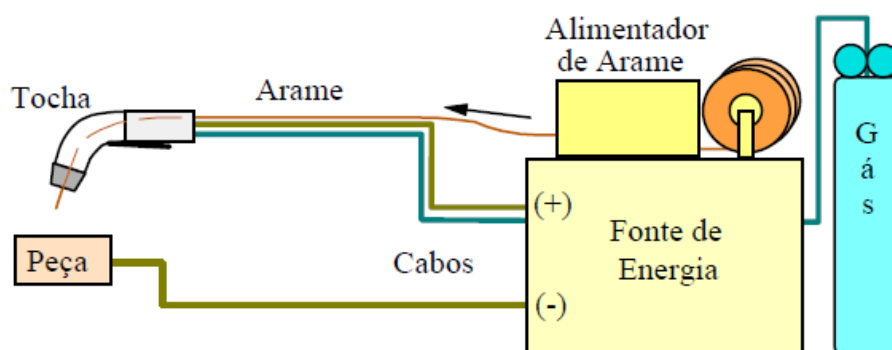


FIGURA 2.36 - Equipamento para soldagem *FCAW*.

FONTE: MODENESI & MARQUES, 2006.

Neste tipo de sistema, a tensão e a velocidade de alimentação do arame permanecem aproximadamente constantes durante a operação de soldagem, pois são ajustados naturalmente através de variações na corrente. Assim, qualquer variação nas condições de soldagem é

absorvida principalmente por alterações na corrente, o que possibilita que a soldagem seja realizada com o comprimento do arco relativamente constante. A fácil abertura do arco e seu comprimento constante durante a soldagem são as principais razões da grande popularidade desta forma de operação.

Os benefícios de sua operação aliados a outras vantagens do processo *FCAW*, como a ampla gama de aplicações e posições de soldagem, altas taxas de deposição, alto fator de ocupação devido ao eletrodo contínuo, fez esse processo ser bastante utilizado na soldagem robotizada.

2.7.1 Robotização do Processo *FCAW*

Quando ocorre a robotização de um processo de soldagem, há uma necessidade em aumentar a repetibilidade, precisão e produtividade, com a mínima intervenção do ser humano, a fim de minimizar defeitos e tornar os cordões de solda mais homogêneos. Durante a execução de uma solda, deseja-se que as características finais do cordão sejam tais que atendam às exigências das normas de qualificação para determinada junta soldada. Isso é obtido reduzindo erros humanos, através da robotização.

Na soldagem *FCAW* robotizada, o movimento da tocha de soldagem é controlado por um manipulador robótico. Após a programação, o equipamento executa operações de soldagem, sem ajuste ou controle por parte do operador de solda. Os parâmetros que são ajustados durante a programação do robô são basicamente:

1. Definição do ponto inicial e final do cordão de solda a ser realizado;
2. Tipo de movimento a ser executado (movimento junta a junta, linear, ou circular);
3. Velocidade de deslocamento da tocha;
4. Ângulo de inclinação da tocha;
5. Distância entre o bico de contato e a peça (D.B.C.P.);
6. Tempo de ativação do arco (atraso entre o comando de abertura do arco e início da movimentação do robô).

O processo *FCAW*, devido à sua elevada produtividade e por permitir alimentação mecanizada do arame, é um processo com alto grau de robotização no meio industrial. Neste processo, todas as funções ou passos de uma operação são executados, em sequência, por meios mecânicos e/ou eletrônicos, sem qualquer ajuste feito pelo soldador, exceto uma eventual programação do equipamento. Assim, após a programação de determinada trajetória de soldagem, o robô executa as tarefas de soldagem, que incluem a abertura e manutenção do arco, deslocamento da tocha e,

por fim, a extinção do arco. A Figura 2.37 apresenta uma célula robótica com configuração básica para soldagem (PINTO, 2011).

Geralmente, os parâmetros tensão de soldagem e velocidade de alimentação do arame são ajustados através do painel de controle do manipulador robótico (FIG. 2.37 (a)), conhecido como *teach pendant*. Apesar destes parâmetros serem ajustados através do painel de controle do robô, seu controle é independente do gabinete de controle, pois este ajuste é feito direto na máquina de solda. Consequentemente, o ajuste desses parâmetros não faz parte da malha de controle.

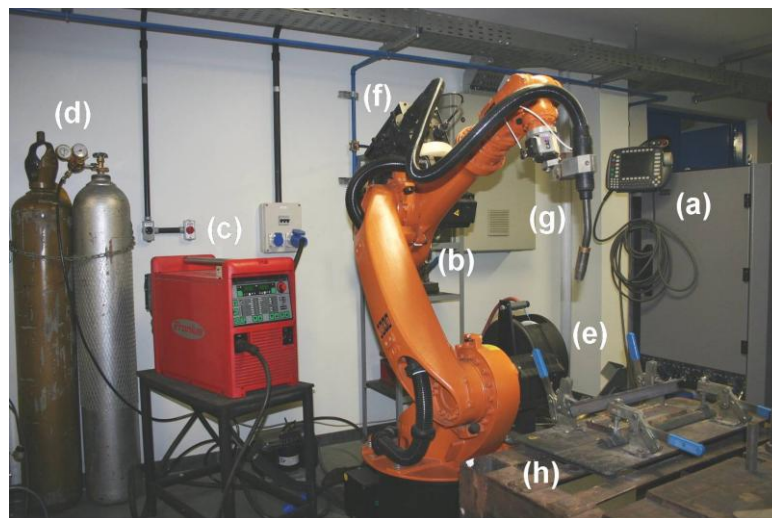


FIGURA 2.37 - Célula robótica de soldagem do LRSS. (a) Painel de controle do robô manipulador; (b) Robô manipulador da Soldagem; (c) fonte de soldagem; (d) cilindro de gás; (e) rolo de arame; (f) alimentador de arame; (g) tocha de soldagem e (h) mesa de soldagem com grampos de fixação.

FONTE: PINTO, 2011.

Algumas vantagens da robotização do processo de soldagem *FCAW* são:

- Solda em diversas posições, com uma escolha apropriada do modo de transferência metálica;
- Trabalha com altas taxas de deposição (kg de metal depositado por hora);
- Permite grandes variações da intensidade da corrente de soldagem, o que torna o processo mais versátil, atendendo diferentes valores de espessuras de chapas;
- Alto ciclo de trabalho devido à continuidade do eletrodo;
- Pequenas variações nos parâmetros de soldagem, em consequência do comprimento do *stick-out* praticamente não variar.

3. METODOLOGIA

A finalidade deste trabalho é executar a remanufatura de um robô utilizando uma arquitetura totalmente aberta, isto é, utilizando uma linguagem de programação não proprietária e um gabinete de controle intercambiável com outros manipuladores robóticos. Desta forma, a metodologia proposta inicialmente foi a execução do *retrofitting* do manipulador ASEA. Seguindo este princípio, componentes de *hardware* como transformadores, motores, diodos, sensores de realimentação e outros dispositivos de potência seriam reaproveitados, fazendo adaptações a novas tecnologias de forma a criar gabinetes de controle universais e de baixo custo (BRACARENSE *et al.*, 2009).

Para manipuladores robóticos industriais que possuem idade avançada, em torno de 35 anos, o processo de *retrofitting* passa a não ser viável, pois os componentes eletrônicos estão bem desgastados e obsoletos. Muitas vezes, o grande problema durante a realização do *retrofitting* é que os componentes, geralmente, não têm um histórico de tempo e condições de uso. Desta forma, há um gasto de tempo e pessoal técnico especializado para avaliar se os componentes poderão ou não ser reaproveitados.

Estudos realizados por Bomfim *et al.* (2012) e Bomfim (2009), comprovaram que durante a realização do *retrofitting* do manipulador ASEA, foi observado que os componentes eletrônicos estavam bastante obsoletos, com consumo excessivo de energia e não apresentavam boa confiabilidade. Devido à incompatibilidade de tecnologia e idade avançada de alguns componentes, chegou-se à conclusão de que seria mais viável a troca de todos os componentes eletrônicos e eletromecânicos reutilizando apenas a mecânica do robô. Com esta mudança de metodologia, o *retrofitting* foi substituído por um novo conceito, denominado *overhauling* (BOMFIM, *et al.*, 2012).

Os manipuladores robóticos com longo tempo de uso, geralmente, apresentam sua estrutura mecânica em boas condições de uso, por se tratar de uma estrutura que é muitas vezes superdimensionada para sua aplicação, no intuito de evitar falhas mecânicas devido ao escoamento do material. Já seus componentes eletrônicos estão bem deteriorados e com tecnologia obsoleta, o que dificulta a realização de parecer técnico favorável ou desfavorável para a reutilização de seus sistemas. Assim, utiliza-se o novo conceito denominado *overhauling*, onde é reutilizada apenas a parte mecânica do robô, uma vez que a troca de todos os

componentes eletrônicos e eletromecânicos muitas vezes não trazem grandes incrementos sobre o custo total do projeto (BOMFIM *et al.*, 2012).

Assim, a nova metodologia proposta para a execução do *overhauling* de manipuladores robóticos industriais segue o fluxograma da Figura 3.1.

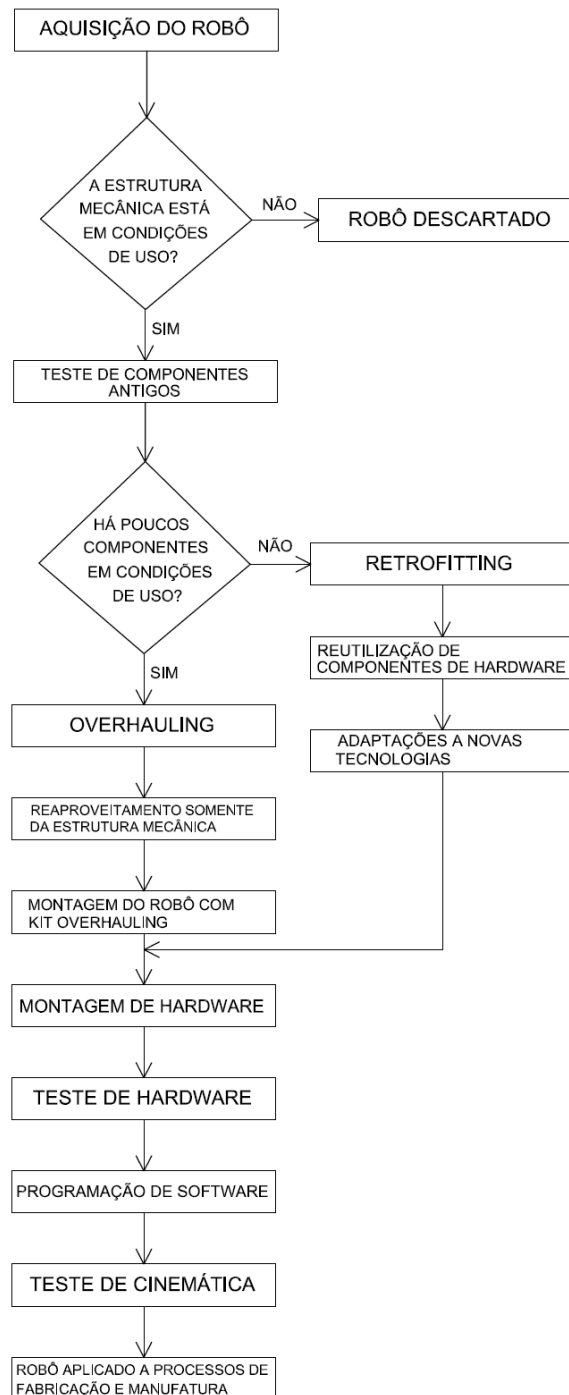


FIGURA 3.1 - Fluxograma com a metodologia proposta para a realização da remanufatura.

Para alcançar o objetivo de realizar o *overhauling* do manipulador ASEA, o presente trabalho foi dividido basicamente em duas etapas, que consistem no projeto e construção de *hardware* e escolha e programação de *software*.

Para projeto e construção do *hardware*, foi necessário seguir sete sub-etapas:

1. seleção dos motores;
2. seleção dos sensores de realimentação;
3. seleção dos *drivers*;
4. projeto e confecção da fonte de alimentação;
5. construção do gabinete de controle;
6. montagem e testes dos motores, sensores de realimentação e sintonia dos controladores;
7. seleção da interface entre o computador e o atuador.

Para a escolha e programação do *software*, foi necessário seguir três sub-etapas:

1. Levantamento do equacionamento geométrico direto e inverso do manipulador robótico (cinemática direta e inversa)
2. seleção do *software* para geração de cinemática;
3. seleção do software para geração de pulsos.

3.1 Projeto e construção do hardware

3.1.1 Seleção dos motores

Os motores utilizados pelo manipulador ASEA antes do processo de remanufatura eram motores de corrente contínua com tensão de alimentação de 35V e potência de 240W. Testes executados nos motores originais verificaram que esses estavam gerando muito ruído, com espiras em curto e consumo excessivo de potência (BOMFIM, 2009). Desta forma, os motores antigos foram substituídos por novos (FIG. 3.2 (a) e (b), respectivamente). Para a seleção dos motores, considerou-se que a potência dos novos deveria ser maior ou igual à potência dos motores antigos. O segundo ponto a ser observado são os acoplamentos que devem ser do mesmo tamanho, para que haja o encaixe perfeito entre o eixo do motor e o eixo da junta de acionamento. Os motores encontrados com o mesmo acoplamento do manipulador ASEA

(NEMA 24) são motores com 350W de potência, o que torna sua utilização viável. Com este acréscimo de 110W de potência, pode-se prever um aumento da capacidade de carga do robô que não pôde ser implementado fisicamente devido a sua estrutura ser projetada para a carga original de 6kg.



(a)

(b)

FIGURA 3.2 - (a) Motor de corrente contínua substituído e (b) motor substituto no processo de *overhauling*.

FONTE: BOMFIM, 2009.

Para verificar se o motor novo estava em perfeitas condições, o mesmo foi acionado em malha aberta, através de uma fonte de alimentação ajustável de 30V, como pode ser visto na Figura 3.3. O potenciômetro da fonte de alimentação foi ajustado suavemente de 0V até 30V, sendo a resposta do motor satisfatória, pois o mesmo não apresentou ruído e oscilação. À medida que a tensão da fonte era aumentada, o motor respondia aumentando sua velocidade angular.

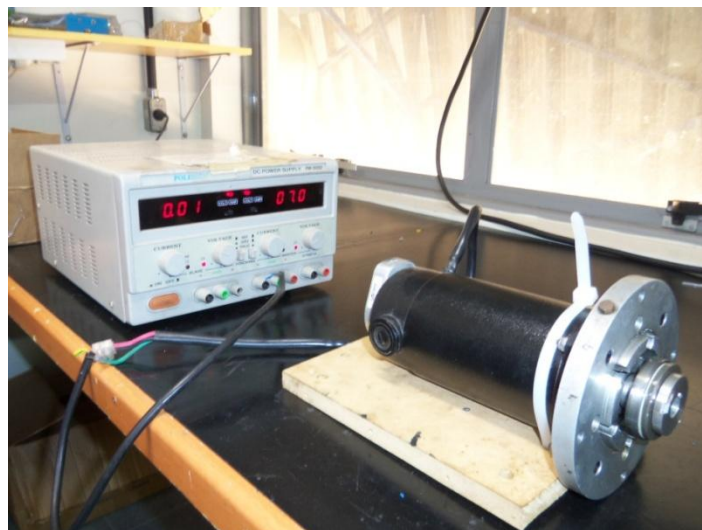


FIGURA 3.3 - Testes do motor operando em malha aberta.

3.1.2 Seleção dos sensores de realimentação

Os sensores de realimentação originais do manipulador ASEA eram compostos por tacômetros e *resolvers*. Desta forma, o posicionamento e velocidade do motor eram mensurados por sensores distintos, o que tornava a realimentação uma tarefa complicada, pois o controlador teria que comparar o sinal de dois sensores antes de enviar o sinal de erro. Para simplificar a realimentação do sinal de posicionamento, velocidade e aceleração, os sensores antigos foram substituídos por *encoders*.

Na Figura 3.4 é apresentado o *encoder* selecionado. Esse sensor foi selecionado por apresentar boa qualidade e por ser amplamente utilizado em sistemas de máquinas CNC. Sua resolução é de 2.048 pulsos por revolução o que fornece uma excelente precisão. Seu módulo de transmissão funciona em quadratura, o que possibilita operação com uma resolução de 8.192 pulsos por revolução (0.044°/pulso).

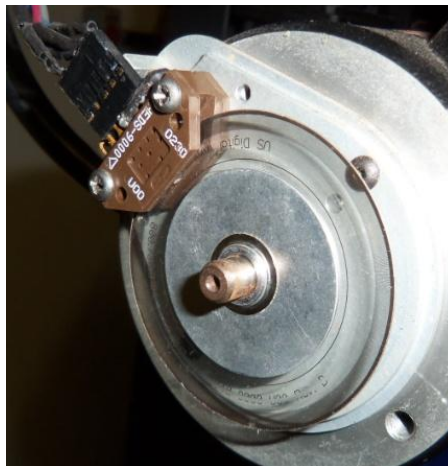


FIGURA 3.4 - *Encoder* fabricado pela USDigital com resolução de 8.192 pulsos por revolução, quando operando em quadratura.

3.1.3 Seleção dos *drivers*

Após a escolha dos motores e dos sensores que fecharam a malha de controle, é feita a seleção dos *drivers*. Os *drivers* devem atender aos requisitos de tensão e potência dos motores e a frequência de comutação deverá ser compatível com o programa gerador de pulsos, a fim de evitar a perda dos mesmos. O *driver* deve ter controlador PID com parâmetros ajustáveis, permitindo assim a sintonia de cada ação de comando, eliminando ruído, oscilações e erros indesejáveis.

Atendendo aos requisitos supramencionados, foi selecionado para o gabinete de controle o *driver* modelo G320 fabricado pela Geckodrive, como pode ser verificado na Figura 3.5. Este *driver* possui uma ampla faixa de tensão (18-80VDC) e corrente (máximo 20A). Essa unidade pode atender a uma ampla gama de motores com potência até 1600W. O atuador possui frequência de comutação máxima de 20kHz e um limitador de corrente externo, além de potenciômetros para ajuste dos parâmetros do controlador PID e uma entrada para parada de emergência. Os motores selecionados para movimentar as juntas do robô possuem tensão de alimentação de 70V e corrente nominal de 5A e os sensores de realimentação são *encoders* com resolução de 2048 pulsos por revolução (resolução de 8.192 pulsos por revolução quando em quadratura). Desta forma, o *driver* selecionado atende satisfatoriamente aos requisitos do sistema.

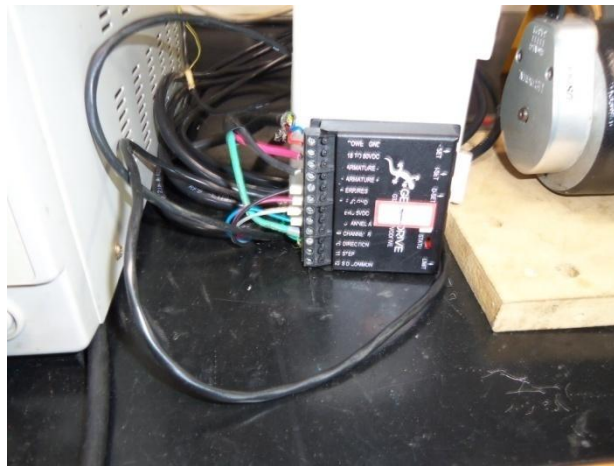


FIGURA 3.5 - *Servodrive* G320 utilizado nos experimentos.

3.1.4 Projeto e confecção da fonte de alimentação

A Figura 3.6 apresenta a fonte de alimentação que deve ser dimensionada de acordo com as necessidades de tensão e corrente dos motores. Tendo em vista que os motores selecionados foram servomotores C.C. de 70V e 5A de corrente (valor nominal) e que o número de juntas é igual a cinco, pode-se calcular a potência total da fonte, a partir da Lei de Ohm, Equação 3.1.

$$Pot = 5 \cdot V \cdot i \quad (3.1)$$

Para o experimento em questão, é dado que: $Pot = 5 \cdot 70V \cdot 5A = 1.750W$, para o caso de corrente nominal.

Para o cálculo correto de potência, a fonte de alimentação deve ser dimensionada levando-se em consideração a corrente de partida dos motores. Desta forma, a potência da fonte teria seu valor em torno de 5.250W, tendo em vista que a corrente de partida dos motores é de 15A. Para o caso

específico deste trabalho, a fonte de alimentação foi projetada para atender à corrente nominal dos motores, pois os mesmos operam em baixas rotações devido a sua finalidade ser a soldagem FCAW.

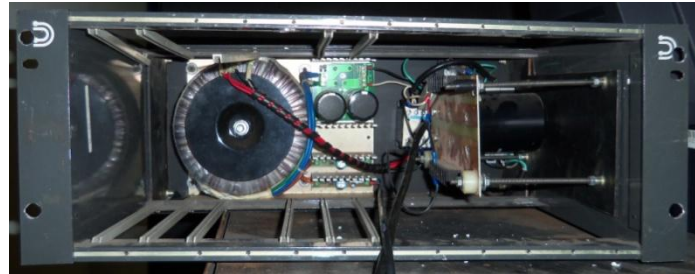


FIGURA 3.6 - Fonte de alimentação de 1.750W utilizada nos experimentos.

3.1.5 Construção do gabinete de controle

A Figura 3.7 apresenta uma visão geral da arquitetura de montagem do gabinete de controle. Basicamente, toda a estrutura de *hardware* fica localizada no gabinete.

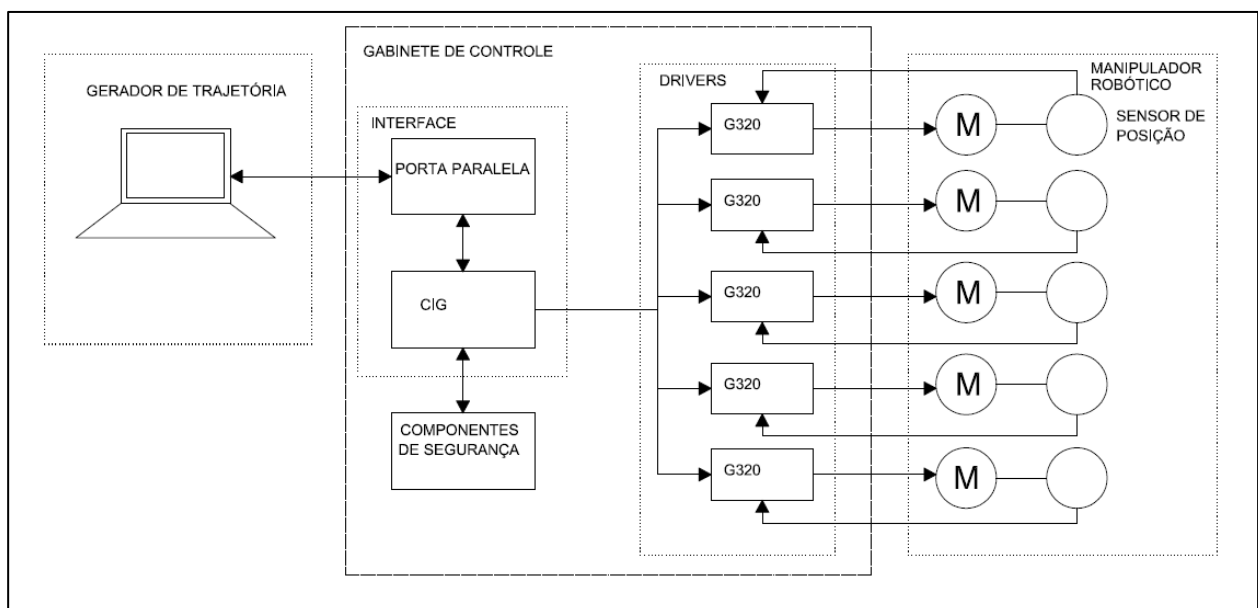


FIGURA 3.7 - Arquitetura de montagem do gabinete de controle.

O primeiro componente eletrônico a ser montado é a fonte de alimentação, pois é o material de maior peso e que exige maior espaço. A seguir é realizada a montagem dos *drivers*. A sequência de ligação dos terminais do *driver* segue as Tabelas 3.1 e 3.2. Os atuadores foram montados sobre uma placa de alumínio (FIG. 3.8) para que ocorra uma maior dissipação de calor, reduzindo a temperatura de operação dos mesmos. A placa Breakout board foi conectada aos *step/direction* de cada atuador, através de fios condutores com coloração diferente para cada *driver*, a fim de facilitar sua identificação. Na base da placa Breakout board, foi colocado um

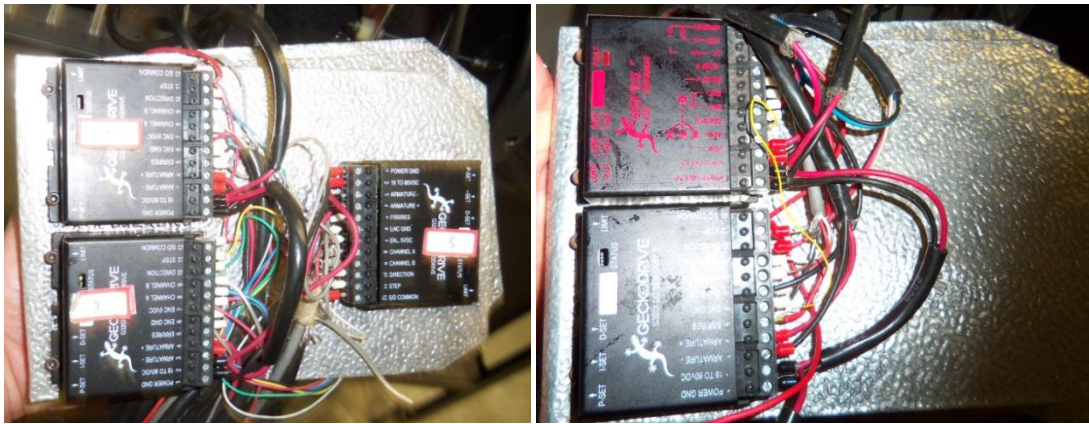
anteparo de isopor com intuito de evitar curto-circuitos com a carcaça do gabinete de controle, como pode ser visto na Figura 3.9.

TABELA 3.1 - Descrição da função de cada pino do *driver* GeckoDrive G 320.

Conector	Descrição	Função
1	POWER GROUND	Ligação do terra (negativo) da fonte de alimentação.
2	+ 18 TO 80 VDC	Ligação do polo positivo da fonte de alimentação.
3	ARM -	Ligação do polo negativo da armadura do motor.
4	ARM +	Ligação do polo positivo da armadura do motor.
5	ERR/RES	Conector de erro/resete. Em curto com o ENC+, aciona o motor. Em curto com o ENC-, desliga o motor.
6	ENC -	Ligação do polo negativo do <i>Encoder</i> .
7	ENC +	Ligação do polo positivo do <i>Encoder</i> .
8	PHASE A	Fase A do <i>Encoder</i> .
9	PHASE B	Fase B do <i>Encoder</i> .
10	DIR	Indica o sentido de rotação do motor.
11	STEP	Indica a quantidade de pulsos que o motor irá movimentar.
12	COMMON	Conectado aos 5 Vdc de uma fonte externa.

TABELA 3.2 - Ligações do *encoder* do motor ao *driver* G320.

Conector	Cor do Fio	Função	Conector do GeckoDrive
1	Preto	Terra	<i>Encoder -</i>
2	Verde	(Index) Contador de rotação	Aterrado (neste experimento a rotação não foi mensurada)
3	Marrom	Canal A	<i>Phase A</i>
4	Vermelho	Alimentação 5Vdc	<i>Encoder +</i>
5	Branco	Canal B	<i>Phase B</i>



(a)

(b)

FIGURA 3.8 - Arranjo de ligação dos *drivers* G320, (a) motores 1, 2 e 3 e (b) motores 4 e 5.

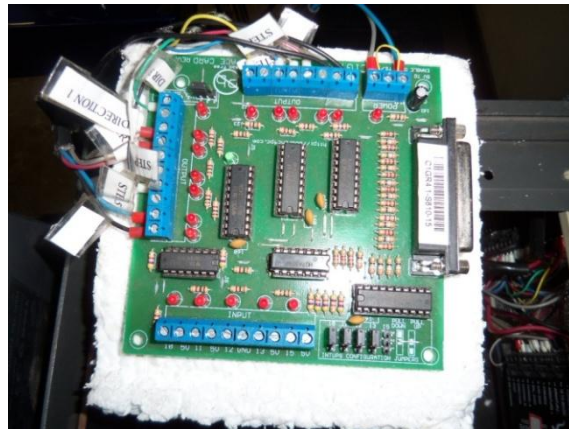


FIGURA 3.9 - Arranjo de ligação da placa modelo C1G - Breakout Board.

3.1.6 Montagem e testes dos motores, sensores de realimentação e sintonia dos controladores

Após a montagem do sistema de hardware do manipulador robótico, uma carga de 6kg foi aplicada ao seu TCP, para simular condições de serviço. O intuito desse teste era verificar se o manipulador ASEA iria suportar a carga para qual foi projetado.

A sintonia dos controladores PID, após a simulação das condições de serviço, teve a finalidade de tornar a resposta do motor mais rápida, com maior estabilidade e com baixo erro em regime permanente.

3.1.7 Seleção da interface entre o computador e o *driver*

Para interligar o computador aos *drivers*, foi utilizada a placa com interface paralela modelo C1G Breakout board produzida pela CNC4 (FIG. 3.10), devido ao fato da porta paralela ter um custo reduzido em comparação com a porta *USB* e não haver a necessidade de configuração e atualização de *firmware*. Outra característica decisiva na escolha deste *hardware* deve-se ao fato

desta placa ser projetada para fornecer uma interface flexível para projetos CNC, utilizando *softwares* de controle de porta paralela, como o Mach 3, por exemplo.

A Breakout Board possui circuitos integrados optoisoladores e bornes de contato. Os circuitos optoisoladores têm a função de isolar o ruído oriundo dos motores e de outros dispositivos como tochas de soldagem, por exemplo. Os bornes de contato têm a função de distribuir os sinais do conector DB-25 e receber sinais para a parada de emergência (EStop) ou acionamento de outros dispositivos, como ferramentas de corte conectadas ao TCP. A cada borne está conectado um diodo emissor de luz (LED) que acende na hipótese de acionamento das entradas *step/direction* de cada *driver*, acionamento da ferramenta de trabalho ou no caso de parada de emergência, facilitando a operação e verificação em casos de falha. Nos bornes, existem doze saídas e seis entradas, o que permite a implantação de até seis motores, isto é, robôs com até seis graus de liberdade e seis sensores de fim de curso e de parada de emergência.



FIGURA 3.10 - Placa com interface paralela modelo C1G-Breakout Board.

FONTE: CNC4, 2012.

3.2 Seleção e programação do *software*

3.2.1 Execução do equacionamento geométrico direto e inverso do manipulador robótico (Cinemática Direta e Inversa do manipulador ASEA)

Quando a cinemática de um robô é programada para uma aplicação específica, o operador define uma trajetória no espaço cartesiano que o TCP deve realizar. Para realizar o movimento pré-definido, cada junta terá que movimentar em ângulos que dependerão das coordenadas pré-estabelecidas (Espaço de Juntas). Para a junta movimentar os ângulos programados, o atuador envia uma quantidade de pulsos, definida por uma relação pulso/grau que modela o movimento de juntas (Espaço de Atuadores). Uma relação deve ser criada entre o Espaço Cartesiano e o Espaço de Atuadores de forma que, ao solicitar determinada posição e orientação do TCP, um

programa determine automaticamente movimentos que cada eixo deve realizar. Esta relação é gerada através da cinemática inversa e cinemática inversa de atuadores.

Alguns trabalhos sugerem a discretização do sistema de coordenadas interno (Espaço de Juntas) e sua correspondência com pontos no sistema de coordenadas externo (Espaço Cartesiano). As desvantagens desses algoritmos são a grande memória necessária para armazenar os espaços discretizados, a necessidade de implementação de algoritmos de busca e a impossibilidade de redução de erros. Além disso, tais métodos são ineficientes para se lidar com singularidades e multiplicidade de soluções (LIMA II, 2005).

Tendo em vista o problema supramencionado, a melhor solução para a implementação da cinemática inversa é o desenvolvimento de equações capazes de calcular a relação entre o Espaço Cartesiano e o Espaço de Atuadores em apenas uma iteração. Assim, a precisão do sistema é melhorada, pois não há a necessidade de armazenar espaços discretizados. A Figura 3.11 apresenta as juntas movimentadas por cada haste e a relação entre o atuador e o respectivo grau de liberdade que é movimentado no manipulador ASEA.

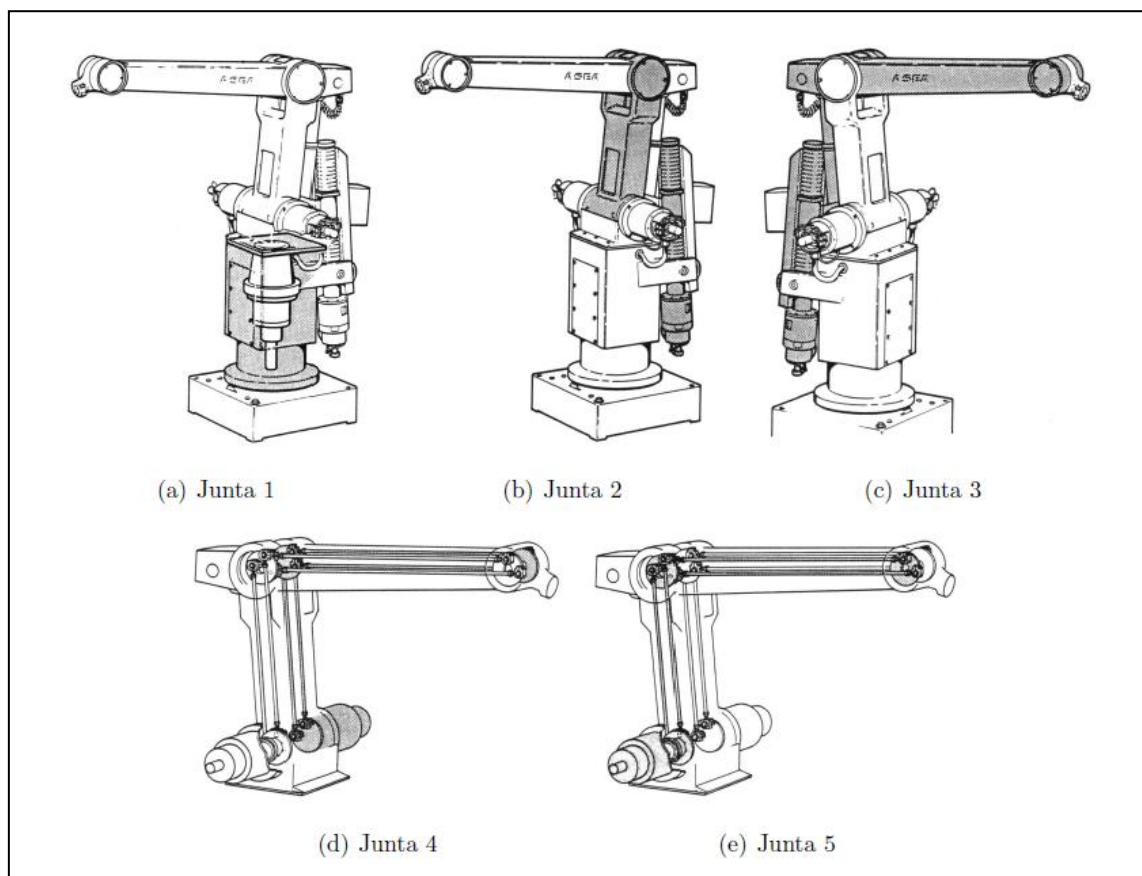


FIGURA 3.11 - Graus de liberdade do Manipulador ASEA.

FONTE: LIMA II, 2005.

Para saber qual é a cinemática inversa de um manipulador robótico, a primeira tarefa a ser feita é calcular a cinemática direta. Após ser calculada a cinemática direta, podem ser realizadas deduções matemáticas de forma a isolar os ângulos das juntas em função da matriz de transformação (cinemática inversa) e isolar o sinal dos atuadores em função dos ângulos das juntas (cinemática inversa de atuadores).

Uma forma simplificada e eficiente de levantar a cinemática direta é através dos parâmetros de Denavit-Hartenberg (SZKODNY, 1995). Para levantar a matriz transformação entre os sistemas de coordenadas do manipulador, seriam necessários seis parâmetros independentes: três de deslocamento do sistema de coordenadas (translação do eixo principal) e outros três de rotação (rotação do eixo principal). Já na notação de Denavit-Hartenberg (D-H), a transformação entre os sistemas de coordenadas de duas juntas adjacentes (sistemas i e $i-1$) é definida somente por quatro parâmetros, a saber:

a_{i-1} = distância de \hat{Z}_{i-1} a \hat{Z}_i medida ao longo de \hat{X}_{i-1} ;

α_{i-1} = ângulo entre \hat{Z}_{i-1} e \hat{Z}_i rotacionado em torno de \hat{X}_{i-1} ;

d_i = distância entre \hat{X}_{i-1} e \hat{X}_i medida ao longo de \hat{Z}_i ;

θ_i = ângulo entre \hat{X}_{i-1} e \hat{X}_i rotacionado em torno de \hat{Z}_i ;

Desta forma, pode-se levantar a Tabela 3.3, onde são identificados os parâmetros de D-H para cada junta do manipulador ASEA.

TABELA 3.3 - Parâmetros de Denavit-Hartenberg para o manipulador ASEA.

Número da junta	α_{i-1} [°]	a_{i-1} [mm]	d_i [mm]	θ_i
1	90	0	700	θ_1
2	0	450	0	θ_2
3	0	670	0	θ_3
4	90	0	0	θ_4
5	0	0	95	θ_5

FONTE: LIMA II, 2005.

De posse desses valores, as matrizes de transformação para cada junta podem ser levantadas a partir da junta anterior. Realizando multiplicações matriciais com as cinco matrizes da Equação 3.2, o resultado obtido será a matriz transformação homogênea que define o TCP do robô em relação a base, como pode ser visto pela Equação 3.3. Nas equações seguintes, a letra S

representa o seno do ângulo e a letra C o cosseno. O índice que acompanha os senos e cossenos representa o ângulo da respectiva junta.

$$\begin{aligned}
{}^0_1T &= \begin{bmatrix} -S_1 & 0 & C_1 & 0 \\ C_1 & 0 & S_1 & 0 \\ 0 & 1 & 0 & 700 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^1_2T &= \begin{bmatrix} -S_2 & -C_2 & 0 & -450S_2 \\ C_2 & -S_2 & 0 & 450S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^2_3T &= \begin{bmatrix} S_3 & C_3 & 0 & 670S_3 \\ -C_3 & S_3 & 0 & -670S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^3_4T &= \begin{bmatrix} -S_4 & 0 & C_4 & 0 \\ C_4 & 0 & S_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^4_5T &= \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & 95 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{3.2}$$

$$\begin{aligned}
{}^0_5T &= {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T = \\
&\begin{bmatrix} S_1S_{234}C_5 + C_1S_5 & -S_1S_{234}S_5 + C_1C_5 & -S_1C_{234} & 450S_1S_2 - 670S_1C_{23} - 95S_1C_{234} \\ -C_1S_{234}C_5 + S_1S_5 & C_1S_{234}S_5 + S_1C_5 & C_1C_{234} & -450C_1S_2 + 670C_1C_{23} + 95C_1C_{234} \\ C_{234}C_5 & -C_{234}S_5 & S_{234} & 700 + 450C_2 + 670S_{23} + 95S_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{3.3}$$

De posse dos ângulos de cada junta e substituindo os valores na Equação 3.3, pode-se obter a matriz transformação que define a posição e orientação do TCP do robô em relação ao sistema de coordenadas da base do robô.

O problema da cinemática inversa é exatamente o contrário: de posse dos valores das coordenadas cartesianas e da orientação da ferramenta, deseja-se calcular os ângulos da junta do robô (LIMA II, 2005). Para o caso específico do manipulador ASEA, será formado um sistema de doze equações e cinco incógnitas (ângulos das cinco juntas), como pode ser visto na Equação 3.4.

$$\begin{aligned}
{}^0_5T &= {}^0_T T {}^T_5 T^{-1} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
&\begin{bmatrix} S_1S_{234}C_5 + C_1S_5 & -S_1S_{234}S_5 + C_1C_5 & -S_1C_{234} & l_2S_1S_2 - l_3S_1C_{23} \\ -C_1S_{234}C_5 + S_1S_5 & C_1S_{234}S_5 + S_1C_5 & C_1C_{234} & -l_2C_1S_2 + l_3C_1C_{23} \\ C_{234}C_5 & -C_{234}S_5 & S_{234} & \lambda_1 + l_2C_2 + l_3S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{3.4}$$

Simplificações matemáticas podem ser realizadas de forma a facilitar os cálculos. O fato da matriz de rotação ser ortonormal faz com que os elementos de sua matriz triangular superior e inferior sejam sempre zero. Assim, pode-se reduzir um sistema de doze equações para seis equações. Outro fator peculiar do manipulador ASEA é que, devido a sua configuração mecânica, a rotação da ferramenta em torno do eixo Z_0 é dependente do valor de θ_1 . Assim, tem-se a Equação 3.5:

$$a_x p_y - a_y p_x = 0 \quad (3.5)$$

Desta forma, o sistema é reduzido para um sistema de cinco equações e cinco incógnitas, cuja solução é dada por Szkodny (1995) na Equação 3.6 (programas em MatLab ANEXO A.2):

$$\theta_1 = \tan^{-1} \frac{-p_x}{p_y}$$

$$\theta_3 = \tan^{-1} \frac{S_3}{C_3}, S_3 = \frac{w_1^2 + w_2^2 - (l_2^2 + l_3^2)}{2l_2 l_3}, C_3 = \sqrt{1 - S_3^2},$$

$$\theta_2 = \tan^{-1} \frac{S_2}{C_2}, S_2 = \frac{w_2 l_3 C_3 - w_1 (l_3 S_3 + l_2)}{l_3^2 C_3^2 + (l_3 S_3 + l_2)^2}, C_2 = \frac{w_1 l_3 C_3 + w_2 (l_3 S_3 + l_2)}{l_3^2 C_3^2 + (l_3 S_3 + l_2)^2}$$

$$w_1 = -S_1 p_x + C_1 p_y + \lambda_5 S_1 a_x - \lambda_5 C_1 a_y,$$

$$w_2 = p_z - \lambda_1 - \lambda_5 a_z$$

$$\theta_{34} = \tan^{-1} \frac{\lambda_5 S_{34}}{\lambda_5 C_{34}}, \quad (3.6)$$

$$\lambda_5 S_{34} = S_1 S_2 p_x - C_1 S_2 p_y + C_2 p_z - \lambda_1 C_2 - l_2 - l_3 C_3$$

$$\lambda_5 C_{34} = -S_1 C_2 p_x + C_1 C_2 p_y + S_2 p_z - \lambda_1 S_2 - l_3 C_3$$

$$\theta_4 = \theta_{34} - \theta_3,$$

$$\theta_5 = \tan^{-1} \frac{S_5}{C_5}, S_5 = C_1 n_x + S_1 n_y, C_5 = C_1 o_x + S_1 o_y$$

Conhecida a cinemática inversa do manipulador é possível definir o ângulo de cada junta a partir da matriz de transformação do TCP da ferramenta. Nos manipuladores robóticos atuais, geralmente, os atuadores agem diretamente nas juntas. Desta forma, pode-se ter uma relação direta (constante K) entre o Espaço de Juntas e o Espaço de Atuadores, no qual esta constante tem relação direta com a resolução do sensor de posição e reduções utilizadas.

O manipulador ASEA foi fabricado em 1977. Nesta época, a preocupação que se tinha era manter os motores de cada junta o mais próximo possível do eixo Z de referência da base. Assim, consegue-se reduzir o momento de inércia provocado pelo peso dos motores, simplificando cálculos de análise estrutural. Em detrimento a esta simplificação, a relação entre o cálculo da cinemática inversa de atuadores tornou-se mais complexa.

No manipulador ASEA, os motores que acionam as juntas quatro e cinco (arfagem e rolamento do TCP do robô) estão localizados no eixo de acionamento da junta dois. Para alcançar as juntas quatro e cinco, foram conectados ao eixo dos motores sistemas de alavancas de forma a transmitir o movimento para as juntas (Figura 3.11). Desta forma, os movimentos das juntas quatro e cinco são dependentes das juntas dois e três, sendo necessário, definir as relações entre o Espaço de Juntas e o Espaço dos Atuadores.

A Tabela 3.4 fornece as dimensões do manipulador ASEA utilizadas para cálculo da cinemática inversa de atuadores.

TABELA 3.4 – Dimensões do manipulador ASEA utilizadas para o cálculo da cinemática inversa de atuadores.

Segmento	Comprimento (mm)
BA	190
BC	240
BD	140
BJ	450
DJ	430
FE	190
FG	240
FH	140

FONTE: LIMA II, 2005.

No primeiro grau de liberdade (giro da base – FIG. 3.11(a)), é dada uma relação direta entre o atuador A_1 e o ângulo da junta θ_1 , através da Equação 3.7:

$$\theta_1 = k_1 A_1 + \lambda_1, \quad (3.7)$$

onde k_1 é a redução da engrenagem (1/158) e λ_1 é uma constante de *offset*.

A segunda junta do manipulador ASEA é acionada através de um fuso linear (segmento DC, da FIG. 3.12). O eixo da junta dois está localizado no ponto B.

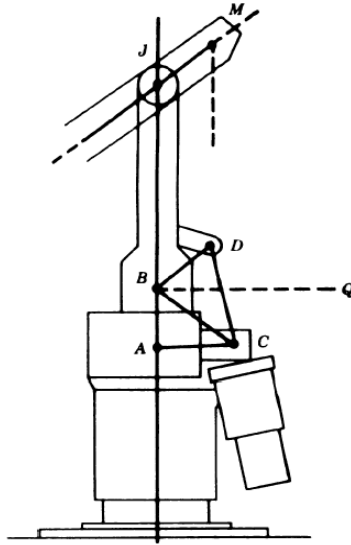


FIGURA 3.12 - Detalhe de acionamento da segunda junta do manipulador ASEA.
 FONTE: LIMA II, 2005.

Através de deduções matemáticas, chega-se a seguinte relação (EQ. 3.8) que define o valor de θ_2 (SZKODNY, 1995):

$$\theta_2 = \cos^{-1} \left(\frac{BC^2 + BD^2 - DC^2}{2 \cdot BC \cdot BD} \right) + \cos^{-1} \left(\frac{BA}{BC} \right) + \cos^{-1} \left(\frac{BJ^2 + BD^2 - DJ^2}{2 \cdot BJ \cdot BD} \right) - 180^\circ \quad (3.8)$$

O comprimento DC é variado através do movimento do motor em um sistema de transmissão que consiste em um fuso. A equação que relaciona o comprimento DC com o movimento do atuador 2 é definido pela Equação 3.9:

$$DC = k_2 A_2 + \lambda_2, \quad (3.9)$$

em que λ_2 é uma constante de *offset*. O passo do fuso é 5mm por volta e a rotação positiva tende a reduzir o comprimento DC. Desta forma, $K_2 = -5\text{mm}/2\pi$ e a Equação 3.9 torna-se a Equação 3.10:

$$\theta_2 = \cos^{-1} \left(\frac{BC^2 + BD^2 - (k_2 A_2 + \lambda_2)^2}{2 \cdot BC \cdot BD} \right) + \cos^{-1} \left(\frac{BA}{BC} \right) + \cos^{-1} \left(\frac{BJ^2 + BD^2 - DJ^2}{2 \cdot BJ \cdot BD} \right) - 180^\circ \quad (3.10)$$

O valor de λ_2 pode ser determinado considerando que $\theta_2 = 0$ e $A_2 = 0$. Desta forma, λ_2 pode ser calculado pela Equação 3.11 da seguinte forma:

$$\lambda_2 = \sqrt{BC^2 + BD^2 - 2 \cdot BC \cdot BD \cdot \cos \left[-\cos^{-1} \left(\frac{BA}{BC} \right) - \cos^{-1} \left(\frac{BJ^2 + BD^2 - DJ^2}{2 \cdot BJ \cdot BD} \right) + 180^\circ \right]} \quad (3.11)$$

Para os comprimentos mostrados na Tabela 3.4, $\lambda_2 = 374,37\text{mm}$.

Da mesma forma que a junta dois, a junta três também é acionada através de um atuador linear (segmento GH, FIG. 3.13). O eixo da junta três está localizado no ponto J.

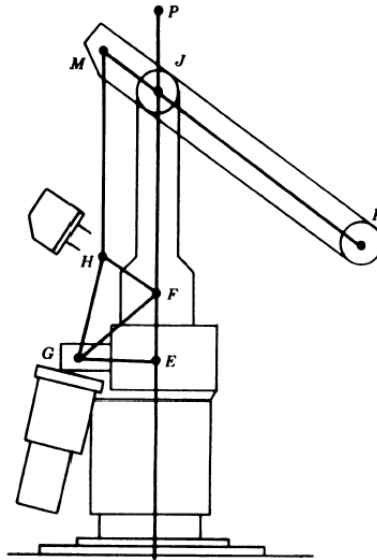


FIGURA 3.13 - Detalhe de acionamento da terceira junta do manipulador ASEA.

FONTE: LIMA II, 2005.

Devido aos aspectos construtivos do manipulador, o movimento da junta três é dependente da junta dois. Desta forma, além de depender do comprimento GH, θ_3 é dependente de θ_2 , como pode ser verificado pela Equação 3.12:

$$\theta_3 = \cos^{-1} \left(\frac{FG^2 + FH^2 - GH^2}{2 \cdot FG \cdot FH} \right) + \cos^{-1} \left(\frac{FE}{FG} \right) - \theta_2 - 90^\circ \quad (3.12)$$

De forma semelhante à junta dois, a junta três também tem um atuador linear (GH), portanto, tem-se a Equação 3.13:

$$GH = k_3 A_3 + \lambda_3, \quad (3.13)$$

desta forma, é obtida a Equação 3.14:

$$\theta_3 = \cos^{-1} \left(\frac{FG^2 + FH^2 - (k_3 A_3 + \lambda_3)^2}{2 \cdot FG \cdot FH} \right) + \cos^{-1} \left(\frac{FE}{FG} \right) - \theta_2 - 90^\circ \quad (3.14).$$

Ambos os atuadores lineares são iguais. Deste modo, $K_3 = -5\text{mm}/2\pi$. O valor de λ_3 pode ser determinado considerando que $\theta_2 = 0$, $\theta_3 = 0$ e $A_3 = 0$. Desta forma, é obtida a Equação 3.15:

$$\lambda_3 = \sqrt{FG^2 + FH^2 - 2 \cdot FG \cdot FH \cdot \cos \left[-\cos^{-1} \left(\frac{FE}{FG} \right) + 90^\circ \right]} \quad (3.15).$$

Para os comprimentos mostrados na Tabela 3.4, $\lambda_3 = 190,1156\text{mm}$.

O quarto grau de liberdade é acionado pelo atuador quatro (A_4), através de um sistema de hastes, como pode ser visto na Figura 3.14.

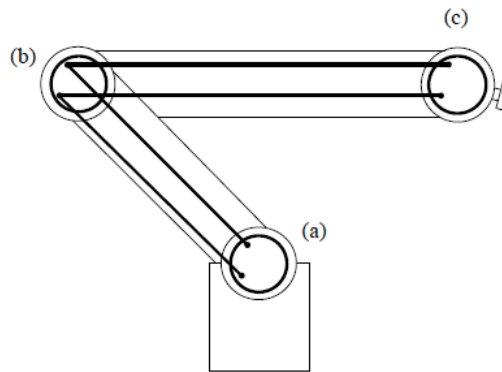


FIGURA 3.14 - Sistema de transmissão por hastes para acionamento da junta 4 do manipulador ASEA.

FONTE: LIMA II, 2005.

Pela Figura 3.14, pode-se verificar que os discos das juntas 2 (a), 3 (b) e 4 (c) possuem o mesmo diâmetro. Desta maneira, o movimento é transmitido com ganho unitário. O ângulo da junta 4 (θ_4) será dependente dos ângulos das juntas 2 e 3 e do movimento do atuador A_4 . O movimento do disco (a) causa um movimento da junta 4 em relação ao sistema de coordenadas da base, como pode ser verificado pela Equação 3.16.

$$\theta_4 = -k_4 A_4 - \theta_2 - \theta_3 + \lambda_4 \quad (3.16)$$

Considerando que um movimento do atuador A_4 no sentido positivo causa uma rotação positiva da junta 4, e que o disco (a) é acionado através de uma engrenagem harmônica de redução igual a 1/128, obteve-se $K_4 = -1/128$. Com as condições de contorno $\theta_2 = \theta_3 = \theta_4 = 0$, pode-se obter o valor de $\lambda_4 = 0$.

De forma semelhante ao quarto grau de liberdade, o quinto eixo também é acionado por um sistema de hastes, conforme a Figura 3.14. A diferença é que o quinto grau utiliza um sistema de engrenagens cônicas para girar de noventa graus o eixo de rotação. Assim, a Equação 3.17 é obtida:

$$\theta_5 = (k_5 A_5 - \theta_4) k_6 + \lambda_5, \quad (3.17)$$

onde $K_6 = 1,5$ é a redução do pinhão e coroa, e $K_5 = -1/128$ é a redução da engrenagem harmônica. Considerando que, para $\theta_4 = \theta_5 = A_5 = 0$, a constante de *offset* (λ_4) também igual a zero.

De forma simplificada, as seguintes equações (EQ 3.18) determinam as posições das juntas em função das posições dos atuadores (cinemática direta de atuadores):

$$\begin{aligned} \theta_1 &= k_1 A_1 + \lambda_1 \\ \theta_2 &= \cos^{-1} \left(\frac{BC^2 + BD^2 - (k_2 A_2 + \lambda_2)^2}{2 \cdot BC \cdot BD} \right) + \cos^{-1} \left(\frac{BA}{BC} \right) + \cos^{-1} \left(\frac{BJ^2 + BD^2 - DJ^2}{2 \cdot BJ \cdot BD} \right) - 180^\circ \\ \theta_3 &= \cos^{-1} \left(\frac{FG^2 + FH^2 - (k_3 A_3 + \lambda_3)^2}{2 \cdot FG \cdot FH} \right) + \cos^{-1} \left(\frac{FE}{FG} \right) - \theta_2 - 90^\circ \\ \theta_4 &= -k_4 A_4 - \theta_2 - \theta_3 + \lambda_4 \\ \theta_5 &= (k_5 A_5 - \theta_4) k_6 + \lambda_5 \end{aligned} \quad (3.18)$$

Invertendo-se as equações anteriores, são dados os valores dos atuadores para determinados valores de ângulos (cinemática inversa de atuadores), como pode ser verificado na Equação 3.19 (programas em MatLab - ANEXO A.2).

$$\begin{aligned} A_1 &= \frac{\theta_1 - \lambda_1}{k_1} \\ A_2 &= \frac{\sqrt{BC^2 + BD^2 - 2 \cdot BC \cdot BD \cdot \cos \alpha} - \lambda_2}{k_2} \\ \alpha &= \theta_2 - \cos^{-1} \left(\frac{BA}{BC} \right) - \cos^{-1} \left(\frac{BJ^2 + BD^2 - DJ^2}{2 \cdot BJ \cdot BD} \right) + 180^\circ \\ A_3 &= \frac{1}{k_3} \left(\sqrt{FG^2 + FH^2 - 2 \cdot FG \cdot FH \cdot \cos \left[\theta_3 + \theta_2 - \cos^{-1} \left(\frac{FE}{FG} \right) + 90^\circ \right]} - \lambda_3 \right) \\ A_4 &= -\frac{\theta_4 + \theta_2 + \theta_3 - \lambda_4}{k_4} \\ A_5 &= \frac{1}{k_5} \left(\frac{\theta_5 - \lambda_5}{k_6} + \theta_4 \right) \end{aligned} \quad (3.19)$$

3.2.2 Seleção do *software* para geração de cinemática

O MatLab (abreviatura de Matrix Laboratory - Laboratório de Matrizes) foi selecionado neste trabalho porque é um *software* de simulação matemática que realiza operações matriciais, constrói gráficos em duas ou três dimensões, auxilia no processamento de sinais, além de manipular outras funções especializadas. Ele trabalha com uma linguagem de programação de alto nível, em um ambiente interativo, para o desenvolvimento de algoritmos, análise e visualização de dados e computação numérica. Próprio para as áreas técnica e científica, o *software* tem funções de tratamento numérico de alto desempenho, capazes de resolver problemas computacionais técnicos de forma mais eficiente do que as tradicionais linguagens de programação. Além do ambiente interativo, outra facilidade do MatLab é a possibilidade de execução de arquivos texto, contendo uma sequência de instruções definidas pelo usuário. Esses arquivos texto, com extensão “.m” podem ser criados e editados dentro ou fora do seu ambiente (CHAPMAN, 2012).

Este *software* é um sistema para cálculo científico que proporciona um ambiente de fácil utilização com uma notação intuitiva. Permite a realização de algoritmos numéricos sobre matrizes com o mínimo de programação. Além disso, ele possui recursos de programação, agindo como uma linguagem semelhante à C, porém voltada para processamento numérico intensivo. O MatLab soluciona problemas diversos tais como: álgebra matricial, aritmética com complexos, sistemas de equações lineares, determinação de autovalores e autovetores, solução de equações diferenciais, solução de equações não lineares, além de representar e de dar subsídios (sub-módulo) para a análise e para síntese de sistemas lineares e não lineares.

Tendo em vista que a cinemática de robôs industriais é descrita através de matrizes de transformação homogênea 4x4, o movimento da ponta de uma ferramenta (TCP) a partir da base do robô seria facilmente descrita utilizando-se o MatLab. Para programar a cinemática do manipulador ASEA em MatLab, é necessário gerar um código fonte da cinemática inversa e outro da cinemática inversa de atuadores. Após gerar esses dois códigos fonte, deve ser gerado um terceiro de forma a determinar os parâmetros da trajetória que se deseja realizar. O problema encontrado para a utilização do MatLab é descobrir como a cinemática encontrada poderá modular as entradas *step/direction* dos *drivers*. Uma solução encontrada foi a utilização do *software* Mach 3.

3.2.3 Seleção do software para geração de pulsos

Para a criação de uma interface amigável entre o operador e a máquina CNC, foram criados *softwares* onde há o controle de cada movimento da ferramenta de corte através de linhas de comandos (*G code*). Dentre estes programas, um amplamente difundido na indústria é o Mach 3. O Mach 3 é um programa muito flexível projetado para controlar máquinas tais como fresas, tornos e cortadores a plasma. Sua função básica é interpretar as linhas de comando do código G e transformá-las em pulsos que irão modular os *drivers*.

O Mach 3 foi criado para o controle de máquinas cartesianas, onde existe uma relação linear entre o Espaço de Atuadores e o Espaço Cartesiano, isto é, o movimento do atuador é diretamente proporcional ao deslocamento na direção cartesiana que ele controla. O Mach 3 possui três versões de página de controle, que são: fresa, torno e máquinas de corte a plasma. No controle de máquinas de corte a plasma e de fresamento, há a possibilidade de controle de posicionamento nos três eixos cartesianos (x, y e z) de forma simultânea. Já nos tornos, há o controle de dois eixos cartesianos (x e z), onde ocorre o controle do volume desbastado em operações de usinagem.

Um das características mais atrativas do *software* Mach 3 para sua utilização neste experimento, é a possibilidade de controlar simultaneamente duas fresas CNC (eixos X, Y, Z, A, B e C da FIG. 3.15), totalizando seis eixos controlados. Se o mecanismo a ser controlado for o braço de um robô, que geralmente possui seis graus de liberdade e um eixo conectado a cada grau, o Mach 3 poderá controlá-los, desde que o operador realize cálculos cinemáticos para relacionar a ferramenta na posição das coordenadas cartesianas ao comprimento e à rotação dos braços do robô.

Dentre as características do Mach 3, podem ser enumeradas as seguintes:

- a conexão entre a máquina e o computador são feitas através de portas paralelas (portas de impressora) ou *USB*;
- as conexões para controle de funções especiais como um display de LCD, um trocador de ferramentas ou um transportador poderão igualmente serem controlados através de um dispositivo de *ModBus* (por exemplo um CLP);
- o Mach 3 pode ligar o motor de uma ferramenta (função *Spindle*, Figura 3.15), girando-o em uma direção ou em outra, podendo igualmente controlar a taxa em que o motor irá girar (rpm) e monitorar sua posição angular para operações com linha de corte. Com esta

função, há a possibilidade de acionar gatilhos de tochas de máquinas de soldagem, por exemplo;

- o *SoftWare* possui interruptores (EStop) que podem funcionar como sensores de fim de curso ou sensores de anti-colisão.

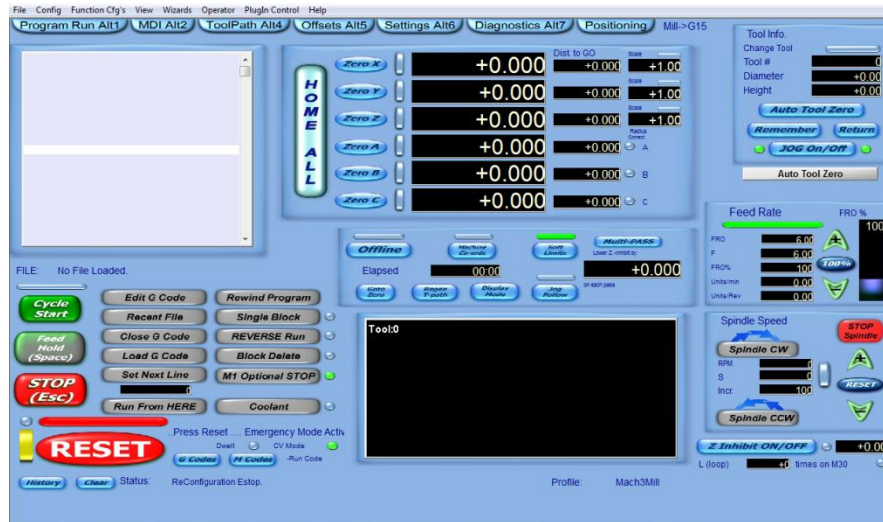


FIGURA 3.15 - Tela de comando do *software* Mach 3 para operar duas fresas simultaneamente.

Com a programação da trajetória através do *software* selecionado, será verificado se o robô atende as características de acionamento, capacidade e precisão de forma satisfatória. A Figura 3.16 apresenta o diagrama esquemático da malha de controle para acionamento de um motor. Apesar de apresentar apenas um motor, a lógica de ligação vale para os cinco atuadores.

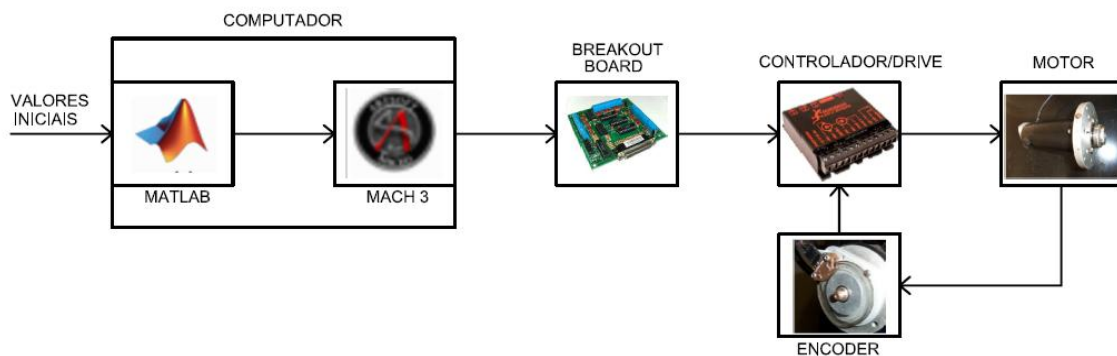


FIGURA 3.16 - Diagrama esquemático da malha de controle de um motor.

3.3 Realização de soldas através do processo de soldagem *FCAW*

Para finalizar a parte experimental, serão realizados cordões de solda através do processo *FCAW*, variando os parâmetros de soldagem, onde serão atribuídas diferentes velocidades de soldagem à tocha, juntamente com diferentes valores de tensão, corrente e tempo de ativação do arco, como pode ser verificado na Tabela 3.5. Os valores de vazão de gás (CO_2 puro) e velocidade de alimentação do arame serão mantidos constantes em 16 litros/minuto e 30 mm/s, respectivamente.

TABELA 3.5 - Parâmetros de soldagem para a realização da parte experimental.

Número de ordem	Velocidade de soldagem (mm/s)	Tensão de soldagem (Volts)	Tamanho de Stick-out (mm)	Tempo de ativação do arco (seg)	Corrente de Soldagem (Ampers)
Experimento 1	20	23	18	0.5	200
Experimento 2	20	25	18	0.1	250
Experimento 3	10	25	18	0.01	324
Experimento 4	15	25	18	0.1	304
Experimento 5	10	33	18	0.1	336
Experimento 6	8	26	18	0.1	320
Experimento 7	8	32	18	0.1	348
Experimento 8	10	33	18	0.1	328

Concluindo o trabalho, serão avaliadas as vantagens e desvantagens do processo de *overhauling*, verificando em quais situações seu emprego é atrativo. Após a conclusão, haverá sugestões para a execução de trabalhos futuros e, no final da dissertação, é apresentado um apêndice com o código fonte dos programas em MatLab utilizados nos experimentos.

3.4 Materiais e equipamentos

TABELA 3.6 - Lista de materiais – *kit-overhauling* do manipulador ASEA.

Item	Descrição	Quant.	Foto	Fabricante
01	Disco de <i>encoder</i> HUBDISK-2 2`` - <i>TRANSMISSIVE ROTARY CODEWHELL</i> – 2048 ciclos por revolução (2048 CPR) – <i>BORE SIZE: 1/4``</i> <i>CODE NUMBER: HUBDISK2 – 2048 –</i> 250 – IE.	05		USDigital
02	Módulo de transmissão do encoder óptico – <i>CODE NUMBER: EM1-2-2048-I</i> – com 02 (dois) canais de quadratura e 2048 CPR de resolução.	05		USDigital
03	<i>Driver</i> modelo G320 Servo Drive. Tensão de operação de 18 a 80V e frequência de operação do sinal <i>PWM</i> de 20kHz.	05		GeckoDrive
04	Conjunto CPU+Monitor Monitor: 14`` de tubo e CPU: processador Pentium Dual Core E 2140 – 1.6GHz, placa mãe GIGA BYTE GA-915GZM-S2, memória 1.0GB DDR2, HD 160 GB SATA e porta paralela.	01		DELL
05	Motor C.C. 70V, 5A. Velocidade máxima 3.300 rpm.	05		keling
06	Breakout board modelo C1G.	01		CNC4



Item	Descrição	Quant.	Foto	Fabricante
07	Fonte de alimentação de 1.750W/70V.	01		-
08	Estrutura mecânica do manipulador ASEA. Ano de fabricação 1977.	01		ASEA

TABELA 3.7 - Lista de *softwares* utilizados no projeto de *overhauling* do manipulador ASEA.



Item	Descrição	Quant.	Foto	Fabricante
01	Software Mach 3 versão Professional 2.0.	01		ArtSoft
02	Software MatLab	01		The MathWorks Inc.

TABELA 3.8 - Lista de equipamentos utilizados no projeto de *overhauling* do manipulador ASEA.


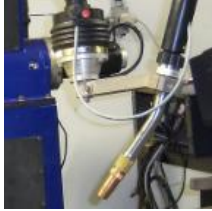


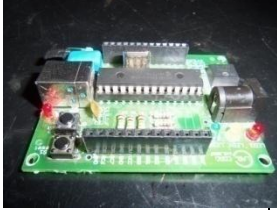

Item	Descrição	Quant.	Foto	Fabricante
01	Máquina para soldagem <i>FCAW</i> modelo Smashweld 408.	01		Esab
02	Conjunto tocha, suporte da tocha e sensor anti-colisão modelo Tbi KS-1.	01		TBi
03	Fonte de Alimentação variável 30V,10A, modelo PMI – 3005D.	01		Polimed
04	Gerador de sinais modelo MFG – 4202.	01		Minipa
05	Placa com microcontrolador PIC18F4550, para geração de 20.000 pulsos em 1.0 segundo	01		Microchip
06	Osciloscópio Agilent, modelo 54622A, com dois canais e precisão de 1.0mV	01		Agilent

TABELA 3.9 - Lista de instrumentos de medição utilizados no projeto de *overhauling* do manipulador ASEA.









Item	Descrição	Quant.	Foto	Fabricante
01	Termômetro infravermelho modelo TI-870.	01		Instrutherm
02	Paquímetro com precisão de 0.05mm.	01		Mitutoyo
03	Alicate amperímetro modelo ET – 3367.	01		Minipa
04	Multímetro modelo 289.	01		Fluke
05	Conjunto potenciômetro multivoltas de 10k Ω (10 voltas) e resistor de 1k Ω	01		-

TABELA 3.10 - Lista de materiais e consumíveis utilizados nos experimentos do projeto de *overhauling* do manipulador ASEA.

Item	Descrição	Quant.	Foto	Fabricante
01	Chapas de aço carbono ASTM A36 de dimensões 6X250.	07		-
02	Chapa de aço API 5LX 80 de dimensões 19X250 com chanfro em meio V e ângulo de 45°.	01		-
03	Consumível: arame tubular ASME SFA-520 E71T-1 com diâmetro de 1.2mm. Especificação Esab: TUBROD71 ULTRA, Lote: 024344.	-	-	Esab
04	Gás de proteção (100% dióxido de carbono).	-		White Martins

4. RESULTADOS E DISCUSSÕES

Nesta etapa do trabalho, serão projetados, confeccionados, programados e testados os circuitos de *software* e *hardware* para execução do *overhauling* do manipulador ASEA. Nas seções a seguir, são apresentadas as características que levaram a utilização de cada componente eletrônico da malha de controle e a motivação do uso dos *softwares* MatLab e Mach 3. Finalizando o capítulo 4, são apresentados diversos cordões de solda realizados através do processo de soldagem *FCAW* para verificação de precisão e repetibilidade do processo de remanufatura.

4.1 Testes de falha do sistema

A Figura 4.1 apresenta a montagem dos testes iniciais, onde foram utilizados os componentes básicos para acionamento de um motor. A sequência correta para acionamento deste dispositivo é ligar a fonte de alimentação e depois fechar um curto entre os pinos 5 e 7 do *driver*, a fim de “resetar” o dispositivo. Após resetar o sistema, foi verificado que o motor apresentava elevado ruído e, ao aplicar um torque em seu eixo, o mesmo entrava em falha ao invés de manter o eixo estático, pois a entrada *step* não estava recebendo dados de entrada. O problema constatado foi que a montagem das fases A e B do *encoder* haviam ficado invertidas. Isso ocasionava muita vibração no motor quando o *hardware* do manipulador ASEA era acionado, além do funcionamento apresentar uma série de falhas.



FIGURA 4.1 - Testes iniciais em bancada.

Quando o dispositivo está ligado de forma correta, ao se aplicar um torque no eixo do motor, o *driver* alimenta sua armadura com uma tensão elétrica contrária e proporcional à força do torque, de forma que o eixo do motor fique estático. Ao alimentar o motor com as fases do *encoder* invertidas, aplicando-se um torque no eixo, o motor tenta manter o eixo imóvel. Porém, com a alimentação invertida nas fases, o motor tende a continuar o movimento aplicado ao invés de aplicar uma força contrária. A explicação para esta falha é que, com as fases invertidas, o *encoder* realimenta um sinal com a fase invertida. Desta forma, para um torque aplicado no sentido horário, o sinal de realimentação tem sentido anti-horário e o controlador do *driver* compara os sinais de comando e realimentação, emitindo um sinal de correção de erro no sentido horário. À medida que o torque aumenta, o erro também aumenta, fazendo com que o controlador entre em falha. Este problema é solucionado quando as fases A e B do *encoder* estão ligadas de forma correta ou há a inversão da polaridade da conexão da armadura do motor.

4.1.1 Dispositivos de segurança

Para maior segurança durante a operação do robô, os *drivers* tiveram seu limite de corrente ajustado para próximo de zero, sendo o cursor do potenciômetro ajustado de zero e girado de 30° em sentido horário, representando um limite de potência de 160W (10% da potência máxima). Desta forma, qualquer colisão da ferramenta de trabalho ou qualquer deslocamento das juntas em ângulos superiores ou inferiores aos limites permitidos por sua estrutura mecânica faz com que os *drivers* entrem em falha facilmente, o que interrompe a operação do sistema.

Outro dispositivo de segurança utilizado é a instalação de duas chaves de segurança na parte traseira do gabinete de controle (FIG. 4.2). Uma chave consiste em um disjuntor bipolar de 30A, que deve ser ligado durante o acionamento da fonte de alimentação. A outra chave é um botão vermelho que deve ser utilizado em caso de emergência. Quando o botão é acionado pelo operador, a fonte de alimentação é desenergizada.

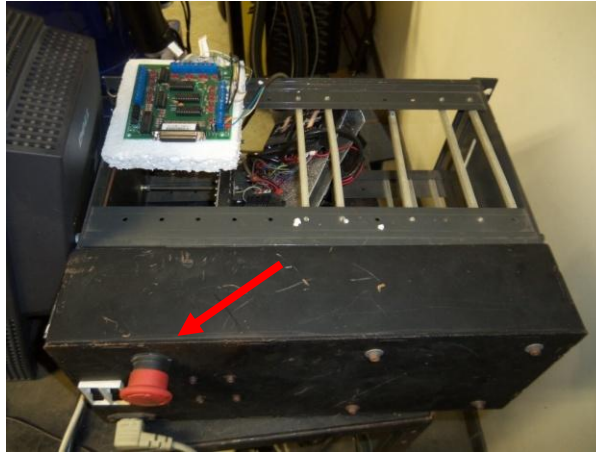


FIGURA 4.2 - Vista de cima do gabinete de controle utilizado nos experimentos, com destaque para o disjuntor e o botão vermelho de emergência.

Podem ser adicionados ao sistema outros dispositivos de segurança que não fazem parte do gabinete de controle. Geralmente, estes dispositivos atuam na placa de interface computador/*drivers* de forma a desligar esta placa. Assim, o sinal de comando não chega até os atuadores. Consequentemente, o robô cessa o seu movimento. Como exemplos desses dispositivos, podem ser citados chaves de fim de curso (*microswitchs*), sensores de presença (fototransistors) e sensores de colisão instalados na ferramenta de trabalho (*microswitchs* e células de carga).

Na Figura 4.3, é apresentado um esquema elétrico com as ligações de *hardware* do gabinete de controle do sistema.

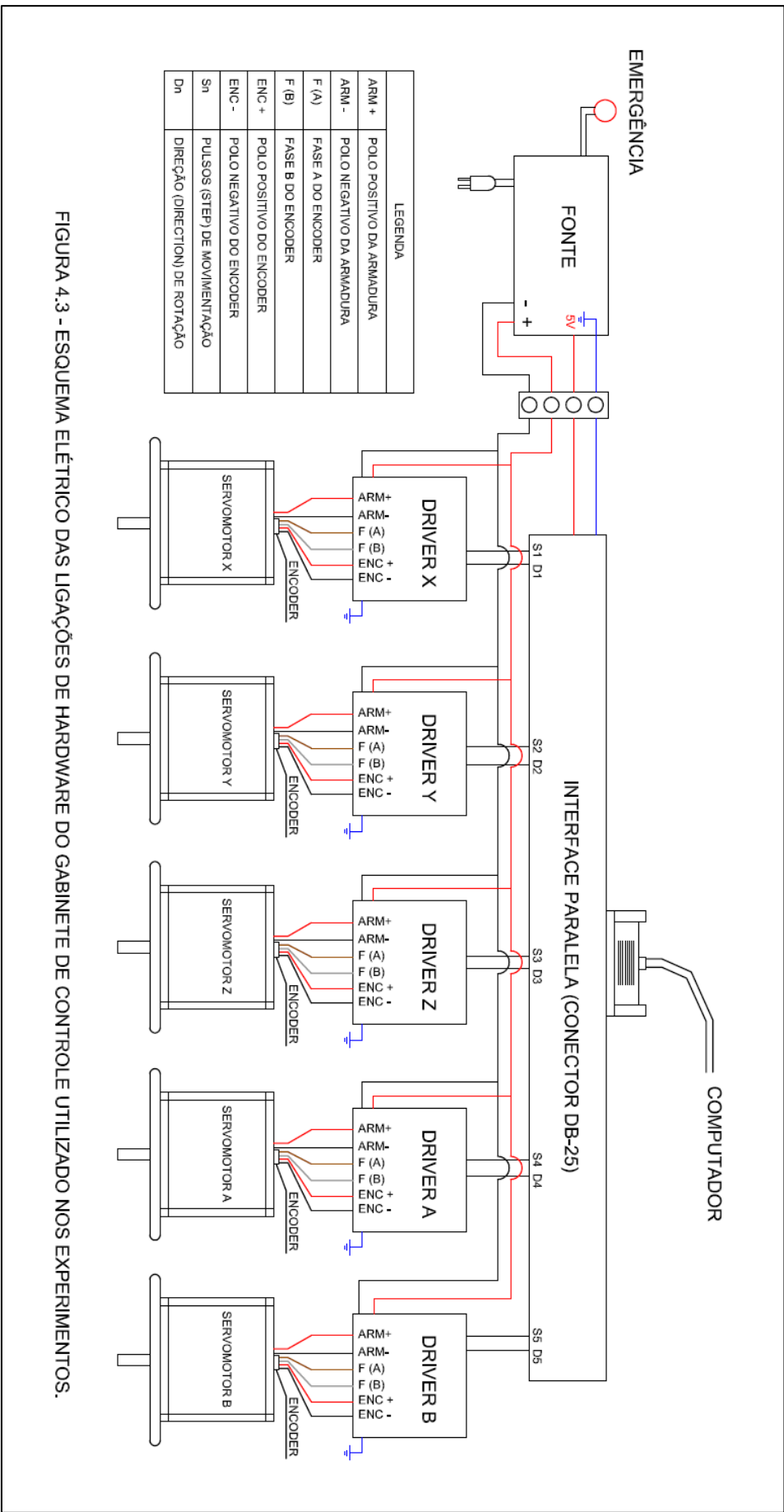


FIGURA 4.3 - ESQUEMA ELÉTRICO DAS LIGAÇÕES DE HARDWARE DO GABINETE DE CONTROLE UTILIZADO NOS EXPERIMENTOS.

4.2 Resposta a perturbações externas

Após a montagem do gabinete de controle, dos motores e sensores de realimentação, a primeira etapa de remanufatura do manipulador ASEA é finalizada, que é o projeto, seleção e montagem do *hardware* de acionamento. Para finalizar esta etapa, é necessário realizar testes para verificar se houve a montagem correta dos componentes eletrônicos, verificação da sintonia dos controladores PID e realização de testes com carga estática máxima.

A característica que o robô deve apresentar após a montagem de todo o dispositivo de *hardware* é ficar estático, mantendo sua posição de referência zero, até que sinais de comando cheguem aos *drivers* e façam ocorrer os movimentos de junta. Todas as perturbações externas que atuam no manipulador ASEA, de forma a tirá-lo da posição de referência, farão com que os *drivers* apliquem uma tensão, com força de mesma intensidade, mas sentido contrário. Após o acionamento do gabinete de controle, o manipulador ASEA teve o comportamento esperado. Perturbações externas foram simuladas através de torques aplicados ao braço do manipulador robótico. O manipulador robótico respondeu de forma satisfatória, mantendo sua posição de equilíbrio estático. O único problema apresentado pelos motores foi uma pequena oscilação e baixo ruído, devido aos controladores PID estarem ainda desregulados. Para simular os torques aplicados, foi utilizado um alteres de 5kg e um saco de areia de 1kg, acoplado ao TCP do robô, como pode ser visto na Figura 4.4.

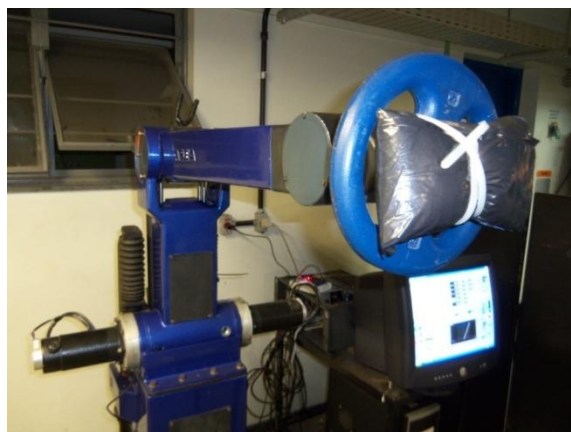


FIGURA 4.4 - Teste de perturbações externas com carga estática de 6 kg.

Nesse experimento os motores não apresentaram aquecimento considerável, pois os *drivers* modulam os motores através de um sinal *PWM*. Desta forma, ocorreu uma variação de temperatura praticamente nula dos motores em relação à temperatura ambiente.

4.3 Sintonia dos controladores PID

A sintonia dos parâmetros das ações de controle proporcional, integral e derivativa se torna uma tarefa complicada neste tipo de planta, pois a equação matemática que modela a função de transferência do manipulador robótico varia de acordo com a posição das juntas. Isso ocorre devido ao fato do robô industrial ser um sistema complexo com acoplamentos dinâmicos entre suas articulações. Em virtude dos efeitos gravitacionais, Coriolis, forças e torque centrífugos, esses acoplamentos são não-lineares.

Essas características fazem com que modelos matemáticos utilizados na literatura, como o Método de Ziegler-Nichols, por exemplo, sejam difíceis de serem implementados fisicamente, pois esses métodos consideram que o comportamento da função de transferência da planta é linear e seus coeficientes são invariantes no tempo. Desta forma, uma solução viável para esse problema seria o projeto de um controlador PID com ajuste adaptativo, onde a sintonia seria refeita para cada posição das juntas e para diferentes valores de carga no TCP do manipulador robótico.

Diante do comportamento dinâmico do manipulador robótico, optou-se pela realização de um ajuste empírico das ações de controle. Esse ajuste consiste, basicamente, no mapeamento das configurações mais severas de operação do robô e no levantamento dos ganhos das ações de controle para cada configuração dos cinco motores. Isso pôde ser realizado devido ao fato do controlador utilizado nos experimentos apresentar elevada robustez, o que foi comprovado empiricamente em outros projetos desenvolvidos no LRSS (PENA, *et al.*, 2011) para variações da equação de transferência da planta.

Para verificar a resposta do motor em malha aberta, foi aplicado um degrau de 2V com a fonte de alimentação. Para a malha fechada, foi utilizada uma placa PIC que gera 20.000 pulsos em um segundo (frequência de 20kHz). Com o intuito de se obter uma resposta gráfica, um potenciômetro multivoltas (10 voltas) de 10 k Ω foi acoplado ao eixo de cada motor (FIG. 4.5) e um osciloscópio foi utilizado para fornecer uma resposta da queda de tensão em um resistor de 1k Ω conectado em série com o potenciômetro. A função do resistor é basicamente limitar o valor da corrente para pequenos valores de resistência do potenciômetro.

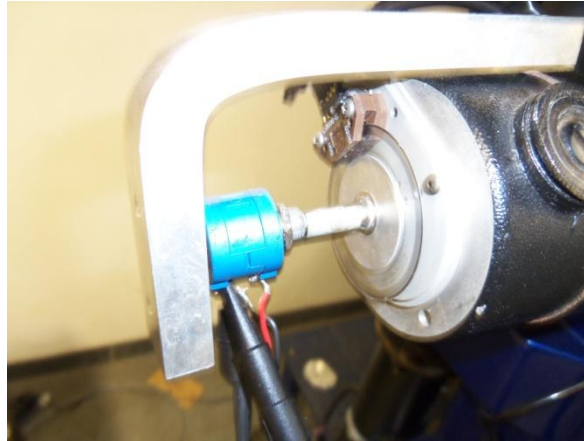


FIGURA 4.5 – Potenciometro multivolts conectado ao eixo do motor 5.

Resposta do motor em malha aberta

Para validar o fato de que a equação de transferência de um manipulador robótico é dinâmica, a resposta ao degrau de 2V para o motor 4 foi levantada para dois tipos de configurações de suas juntas, o que pode ser verificado na Figura 4.6 (a) e (b). A diferença de potencial de 2V foi selecionada em consequência do robô responder com uma velocidade que gera uma curva satisfatória com a utilização do potenciômetro selecionado.

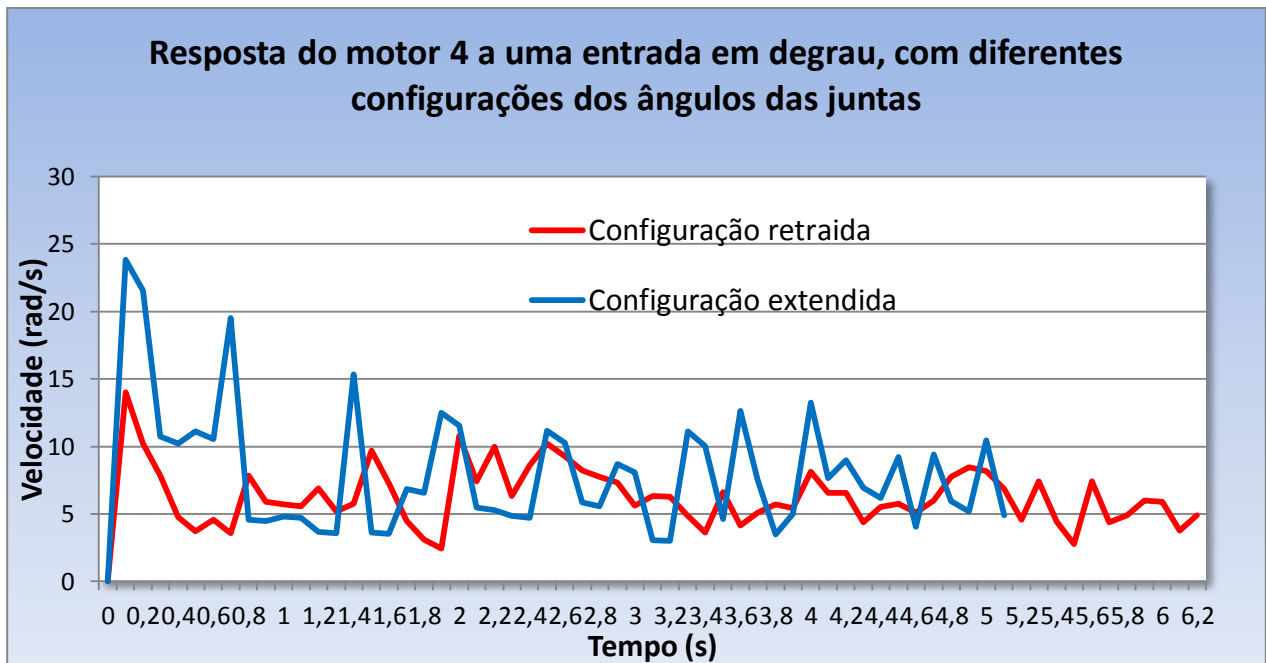
Analisando as duas respostas em malha aberta, pode-se concluir que as equações de transferência que regem cada configuração são distintas, pois, para um mesmo sinal de entrada aplicado (degrau de 2V), a resposta do motor foi diferente nas duas situações. A resposta em malha aberta do motor 4, para o caso do robô retraído, é mais lenta e possui uma taxa de inclinação menor (FIG. 4.6 (c)), o que confirma a diferença entre as equações de transferência entre as duas configurações. Essa primeira análise é qualitativa, com a finalidade de apenas comprovar que a função de transferência que representa o sistema varia com a modificação das configurações dos ângulos das juntas do manipulador ASEA.



(a)



(b)



(c)

FIGURA 4.6 – Resposta em malha aberta do motor 4 para as configurações (a) robô retraído e (b) robô estendido e (c) resposta ao degrau de 2V.

Resposta do motor em malha fechada

Após levantar as curvas em malha aberta dos motores do manipulador ASEA, a malha de controle do sistema foi fechada utilizando os *encoders*. Para melhor desempenho do processo, iniciou-se a sintonia das ações de controle proporcional, integral e derivativa. Devido a fatores supramencionados, optou-se por realizar uma sintonia empírica baseada em orientações do fabricante, na finalidade de cada ação de controle e em seus circuitos digitais equivalentes (TAB. 2.1).

A sintonia das ações de controle é feita através de potenciômetros (FIG. 4.7) e o fabricante não fornece uma relação entre os ganhos K_p , K_i e K_d e o giro do potenciômetro de cada ação de controle. Desta forma, o giro do cursor foi discretizado em nove posições, que variam da posição zero (cursor em zero graus) até a posição nove (cursor em 270 graus ou no fim de curso do potenciômetro). As posições 1, 2, 3, 4, 5, 6, 7 e 8, correspondem ao giro de 30, 60, 90, 120, 150, 180, 210 e 240 graus, respectivamente.

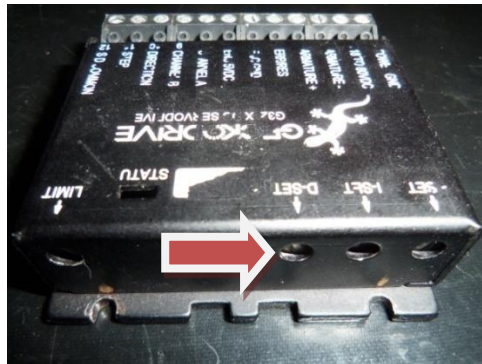


FIGURA 4.7 – *Driver* evidenciando os três potenciômetros das ações de controle PID.

Com o potenciômetro selecionado, existe uma relação direta entre o giro do cursor e o giro do motor. Devido ao fato da placa PIC ser programada para gerar 20.000 pulsos em um segundo, a cada envio de sinal, o motor terá que deslocar $4,88\pi$ radianos ou 2,44 voltas. Esse valor de voltas pode ser obtido dividindo-se a quantidade de pulsos gerados pela resolução do *encoder*, ou seja $20.000/8.192 = 2,44$. Esse valor de voltas implica em uma variação de $2,44k\Omega$ no valor da resistência do potenciômetro (cada volta corresponde a uma variação de resistência de $1,00k\Omega$). Consequentemente, através da queda de tensão no resistor, pode-se obter o valor do deslocamento total do motor, bem como o aspecto de sua curva de resposta, como será visto mais à frente.

A técnica utilizada para sintonia foi o ajuste das ações de controle para as condições mais severas de operação, com carga no TCP máxima. Assim, optou-se pela sintonia dos parâmetros P, I e D nas situações onde as juntas sofrem maiores solicitações. Quanto maior a solicitação de um eixo, mais lenta é a sua resposta em malha aberta. Desta forma, chegou-se a conclusão que a configuração mais severa de operação para os motores 2, 3, 4, e 5 é a posição do braço do robô retraído e para o motor 1 é a posição do braço estendido (vide Figura 4.6). Os ganhos estabelecidos serão utilizados nas demais configurações do manipulador robótico. Cabe ressaltar que um controlador adaptativo não foi proposto neste trabalho, devido ao fato do fabricante do

driver não fornecer os dados sobre os circuitos eletrônicos internos, o que torna inviável um controle digital externo.

Ajuste dos potenciômetros de cada ação de controle

Para o ajuste das ações de controle foram seguidas as seguintes etapas:

1ª etapa: todos os potenciômetros devem ser zerados, isto é, ajustados no sentido anti-horário até o início de curso.

2ª etapa: sintonia da ação de controle proporcional (P).

Justificativa e análise dos resultados obtidos: em sistemas de controle, a ação P é sempre a primeira a ser sintonizada, pois as demais ações de controle atuam no erro gerado por esta ação. O ganho proporcional foi ajustado de 1 a 3, situação na qual o sistema apresentou rigidez satisfatória. Para valores acima de 5, o controlador entra em falha, pois esse ganho representa o ganho crítico do sistema (K_c), a partir do qual a oscilação é bastante elevada. Para simplificar os resultados, a sintonia aplicada será representada na forma P/I/D. Para o caso em questão, a sintonia é 3/0/0.

A Figura 4.8 apresenta a curva de resposta do motor 5, onde o erro residual foi de 5,5% e uma pequena oscilação foi verificada, quando o motor inicia o repouso.

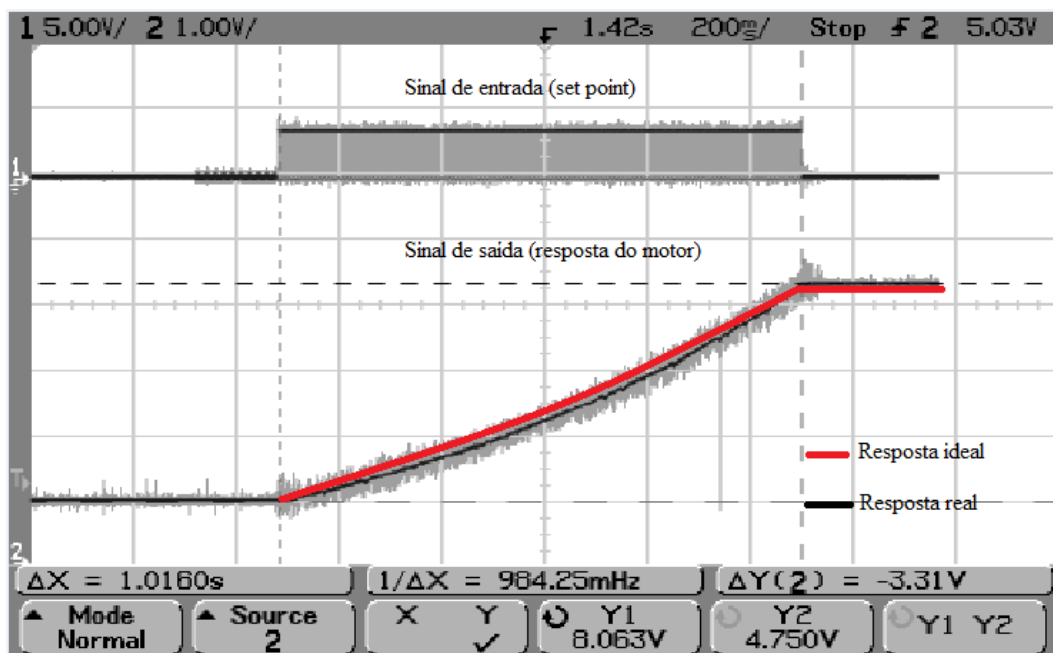


FIGURA 4.8 – Sinal de entrada gerado pela placa PIC (canal 1) e resposta do motor verificada através da queda de tensão no potenciômetro (canal 2) para uma sintonia 3/0/0.

3ª etapa: sintonia da ação de controle derivativa (D)

Justificativa e análise dos resultados obtidos: a ação D prevê o erro atuante, inicia uma ação corretiva e aumenta a estabilidade do sistema, o que permite o uso de valores mais elevados de K_p e K_i , resultando em maior precisão em regime permanente. O ajuste selecionado para a ação D foi o valor 1. Esse valor foi selecionado, porque valores superiores a 1 não apresentaram melhorias na resposta do sistema. Assim, a sintonia adotada foi a 3/0/1.

A Figura 4.9 apresenta a curva de resposta do motor 5, onde o erro residual continuou em 5,5% e a oscilação do sistema é atenuada quando o motor inicia o repouso, representando uma oscilação quase nula.

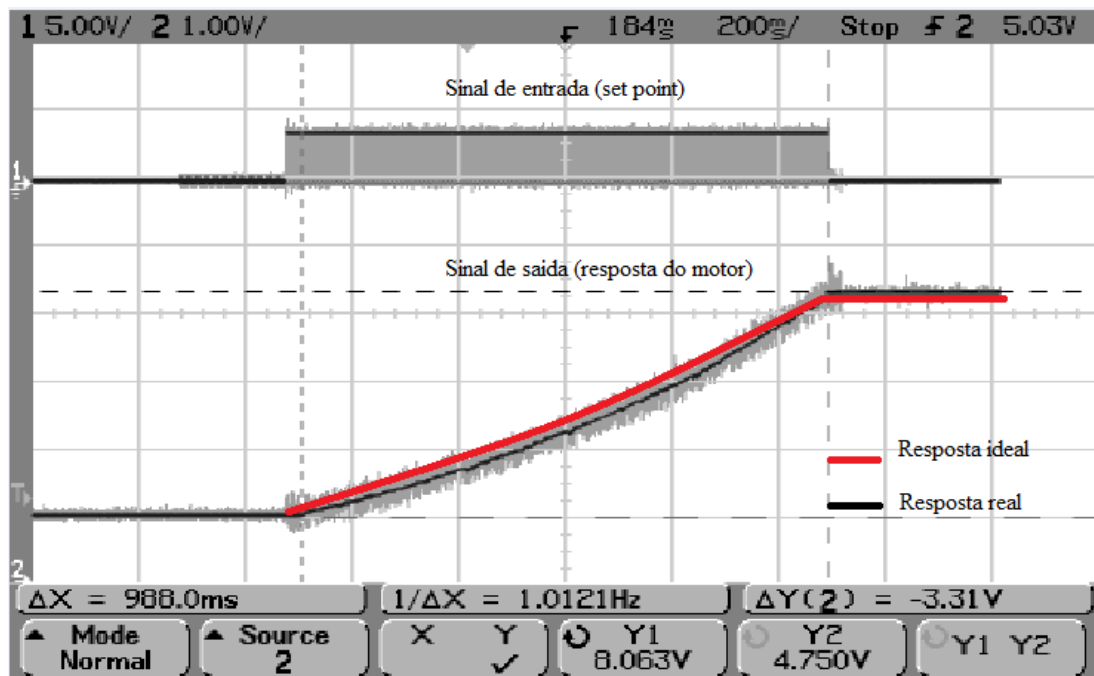


FIGURA 4.9 – Sinal de entrada gerado pela placa PIC (canal 1) e resposta do motor verificada através da queda de tensão no potenciômetro (canal 2) para uma sintonia 3/0/1.

4ª etapa: sintonia da ação de controle integrativa (I)

Justificativa e análise dos resultados obtidos: o erro residual do sistema está em torno de 5,5%. Consequentemente, a ação I foi ajustada para o valor 1, o que resultou em um erro estacionário de 2,4%. Na sequência, a ação I foi ajustada para 2, resultando em um erro estacionário de 0,4%. Esse valor de sintonia foi adotado em consequência de valores superiores a 2 não fazerem efeito na redução do erro. Para valores de sintonia superiores a 6, o sistema entra em instabilidade, o que pôde ser verificado com um giro abrupto do eixo 5.

Pela Figura 4.10, pode ser verificado que o resultado desta configuração de sintonia foi bastante satisfatório, pois para uma sintonia 3/2/1 o erro em regime estacionário foi de 0,4% e a oscilação da resposta foi quase nula.

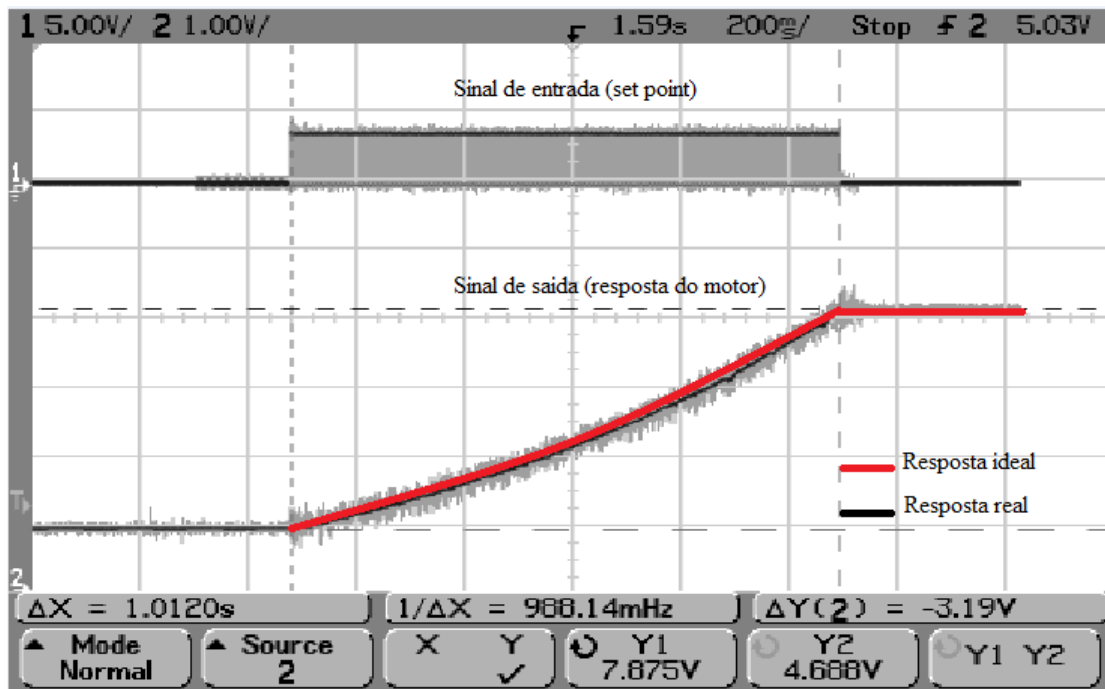


FIGURA 4.10 – Sinal de entrada gerado pela placa PIC (canal 1) e resposta do motor verificada através da queda de tensão no potenciômetro (canal 2) para uma sintonia 3/2/1.

A Figura 4.11 apresenta as respostas do motor 5 para as diferentes sintonias propostas neste trabalho. Pela evolução das curvas, pode-se ter uma noção qualitativa da finalidade de cada ação de controle no sistema.

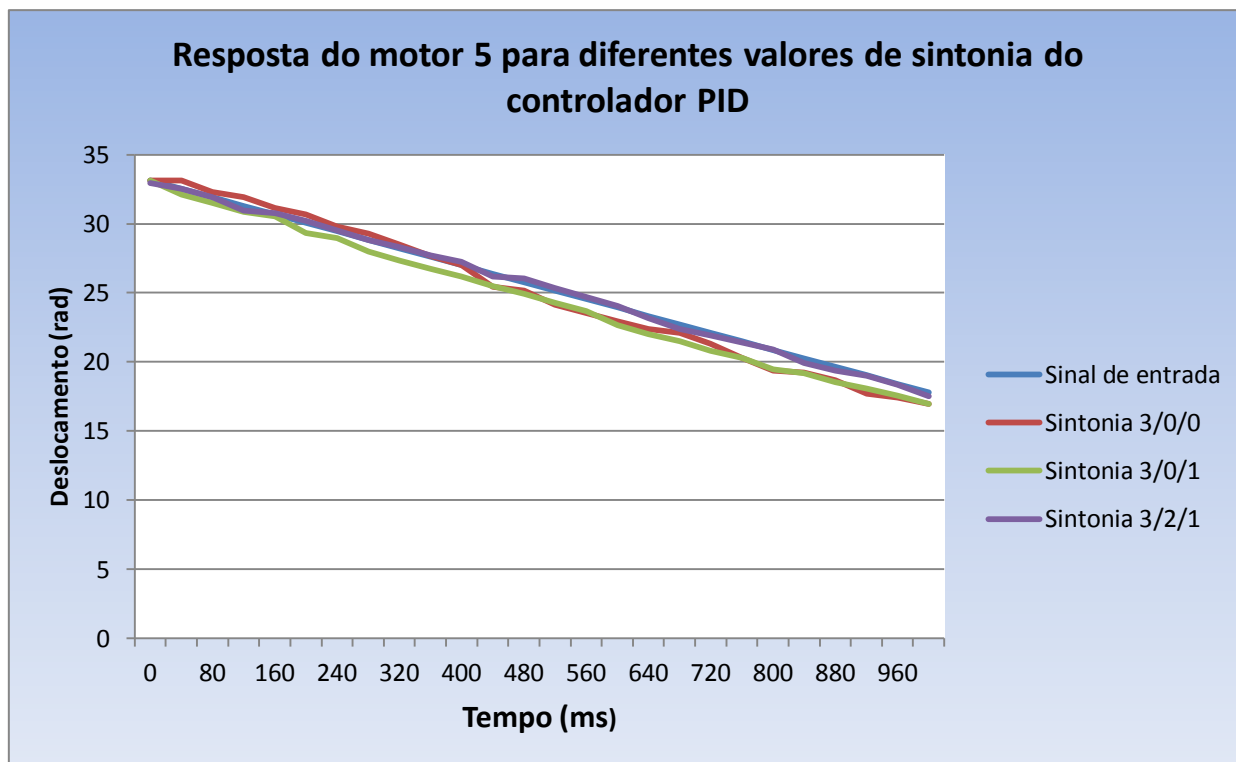
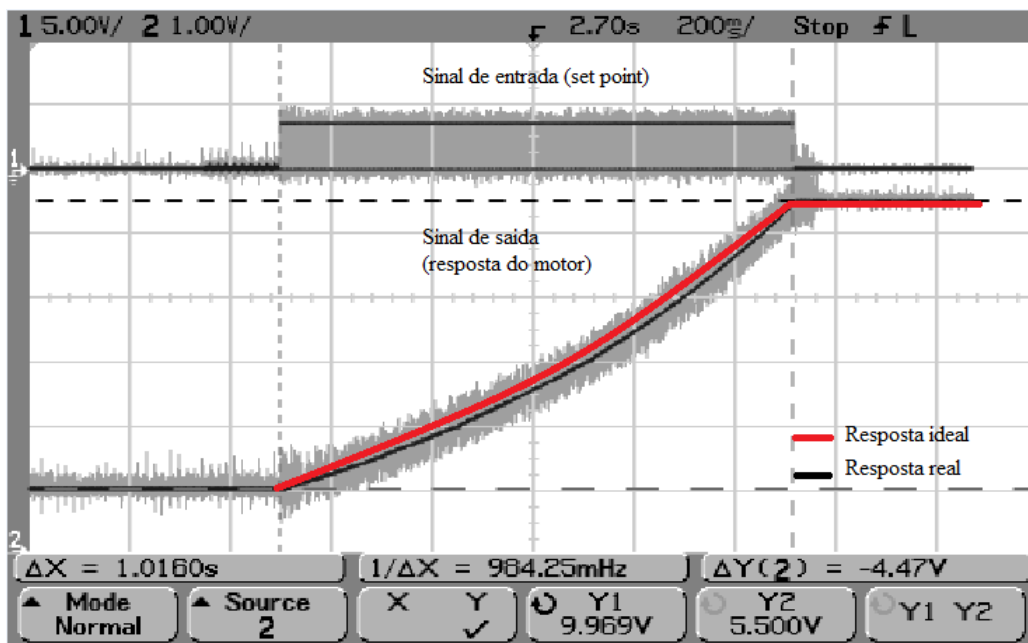


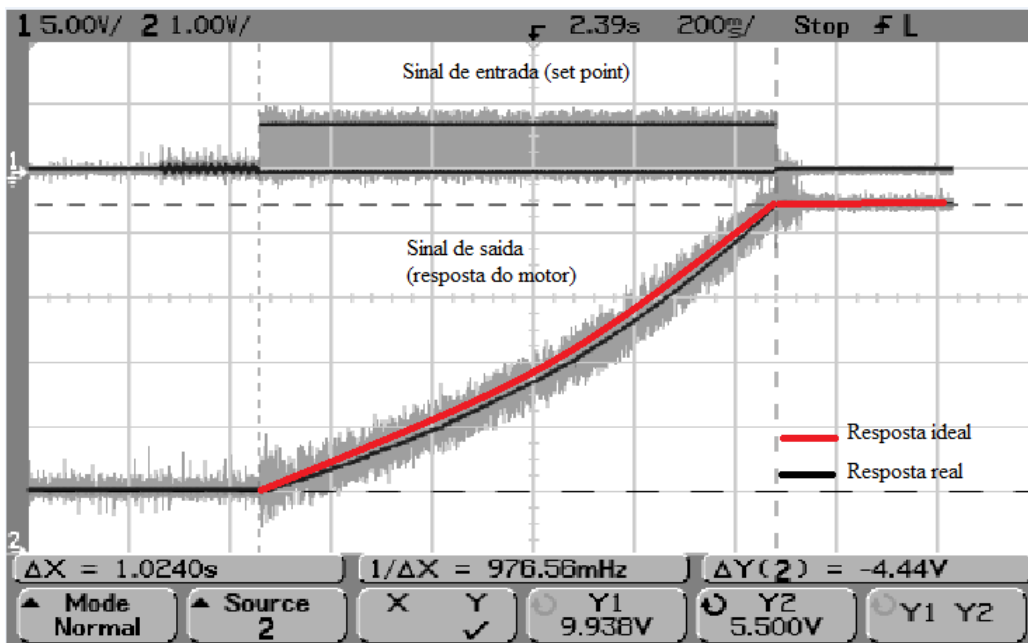
FIGURA 4.11 – Resposta do motor 5 para uma entrada em rampa, com diferentes sintonias do controlador PID.

Pela Figura 4.11, pode ser verificado que a primeira sintonia realizada (3/0/0) apresenta elevada oscilação e elevado erro residual. Com a sintonia 3/0/1, ocorreu um amortecimento da resposta do sistema, atribuído à ação de controle derivativa. Pela figura, pode ser analisado que a ação derivativa não afeta o erro estacionário, pois as curvas de sintonias 3/0/0 e 3/0/1 possuem a mesma divergência em erro, quando comparadas ao *set point* (sinal de entrada). Para atenuar ou até mesmo zerar o erro residual, a ação de controle integrativa foi sintonizada (sintonia 3/2/1) e o erro em deslocamento foi praticamente eliminado, o que pode ser visto pelo encontro das curvas do sinal de entrada e resposta do motor com sintonia 3/2/1 do controlador.

Para verificar o comportamento da sintonia proposta para o motor 5, foram levantadas as curvas de resposta para diferentes configurações do manipulador ASEA e diferentes valores de carga. A Figura 4.12 (a) apresenta a resposta para a configuração estendida com carga de 6kg. Nesta configuração, o erro residual foi de 0,96%. A Figura 4.12 (b) apresenta a resposta do motor para a configuração zero de referência da máquina e TCP sem carga. O erro residual foi de 0,58%. Em ambas as situações, a resposta do controlador foi satisfatória, pois o erro estacionário foi menor que 1,0% e o motor não apresentou grandes oscilações na resposta em posição, para a entrada em rampa.



(a)



(b)

FIGURA 4.12 – Comparação da resposta em malha fechada para o motor 5, com sintonia 3/2/1, para diferentes configurações das juntas: (a) resposta do motor na configuração estendida com carga de 6kg e (b) resposta do motor na configuração zero de referência sem carga.

De forma satisfatória, para o experimento em questão, o controlador foi sintonizado para o pior caso, configuração retraída, e um desempenho pior foi aceito em outros pontos de operação. A

Tabela 4.1 apresenta a sintonia dos controladores PID e o erro residual para cada motor. O giro do potenciômetro de cada ação de controle deve ser realizado no sentido horário, a partir do ajuste zero.

TABELA 4.1 - Ajuste dos parâmetros do controlador PID e erro residual para cada motor.

Ação de controle	Ganho do potenciômetro do motor 1	Ganho do potenciômetro do motor 2	Ganho do potenciômetro do motor 3	Ganho do potenciômetro do motor 4	Ganho do potenciômetro do motor 5
Proporcional	2	3	3	3	3
Integral	1	1	1	2	2
Derivativa	1	1	1	1	1
Erro residual	0,6 %	0,3%	0,3%	0,3%	0,4%

4.4 Limites de utilização de *hardware* e *software*

Durante a programação da cinemática do robô, existem alguns limites que devem ser respeitados no que diz respeito à capacidade de acionamento ou à frequência máxima de operação. Nas próximas seções, serão definidos os limites de operação dos componentes de *hardware* e dos programas de *software* utilizados neste trabalho.

4.4.1 Fonte de alimentação

A potência da fonte de alimentação utilizada é de 1.750W e sua tensão é de 70V. A fonte de alimentação é projetada para alimentar cinco motores de 5A e 70V. Nada impede que a fonte seja utilizada em aplicações onde os motores possuem corrente de alimentação diferente, desde que a tensão dos motores seja 70V e que a soma das correntes dos motores não ultrapasse 25A.

4.4.2 Drivers e encoders

Na especificação do *driver*, existem três fatores principais que devem ser analisados durante a sua seleção:

- a potência máxima de operação;
- a frequência máxima do sinal *PWM*;
- a tensão de operação.

No *driver* selecionado para este experimento, a potência máxima de operação é de 1.600W e sua tensão varia de 18 a 80V, o que atende aos motores selecionados. A frequência máxima de seu sinal *PWM* é de 20kHz. Isto quer dizer que, se forem aplicados pulsos com uma taxa superior a 20.000 por segundo, o *driver* entra em falha.

No experimento, foram utilizados *encoders* com resolução muito elevada (8.192 pulsos por revolução) que, somado às características mecânicas do manipulador ASEA (reduções harmônicas de até 1/158 – engrenagem harmônica da base), fizeram com que a velocidade do TCP do robô ficasse limitada a 55mm/s com movimento linear. Por exemplo, se fosse reduzida a precisão do sistema trocando os *encoders* de 8.192 pulsos por revolução por *encoders* de 400 pulsos por revolução, a velocidade do robô seria aumentada em vinte vezes, aproximando a valores que podem chegar a 1.100mm/s.

4.4.3 Interface computador/*driver*

Existem basicamente três formas de fazer a interface entre computador e *driver*. Através de uma porta paralela, interface *USB* ou conexão com cabo ethernet (par trançado). As vantagens que levaram à utilização da porta paralela devem-se ao fato de ser uma interface barata, quando comparada às demais, e por não necessitar de instalação de *firmware*. Apesar da instalação da porta paralela ser de custo reduzido e de mais fácil execução, existe a desvantagem desta interface possuir um limite de transferência de dados de 3MBps (aproximadamente 60kHz por eixo controlado). Esta característica não foi limitante na utilização desta interface no experimento, pois o *driver* limita a frequência em um valor inferior de operação (20kHz), como foi explicado anteriormente.

4.4.4 Mach 3

O *software* Mach 3 foi criado para utilização no sistema operacional Windows. Como geralmente ocorre em *softwares* de Comando Numérico Computadorizado, o Mach 3 gera pulsos que irão modular motores para aplicações específicas. A frequência máxima com que estes pulsos são gerados é comandado pelo Kernel do Mach 3 (velocidade de Kernel = velocidade de geração de pulsos). Durante a realização dos experimentos, o valor de Kernel utilizado foi de 25kHz por ser uma frequência próxima aos limites de utilização dos *drivers*.

4.4.5 Limites de velocidade do manipulador ASEA

O comando *Feedrate* do código G indica a taxa de avanço da ferramenta de corte. Esta taxa de avanço é padronizada pelo sistema ISO e sua unidade de medida é mm/min. Durante a fabricação de uma máquina CNC, o projetista deve construir o equipamento de forma que obedeça à norma ISO, isto é, quando o operador da máquina programar um *Feedrate* de 10.000 (F10000), a ferramenta terá que deslocar com uma velocidade de 10.000mm/min. Desta forma, o projetista terá que calcular a relação entre resolução de sensores de realimentação, relações de engrenagens das juntas cartesianas e taxa de pulsos a ser fornecida para que o resultado final seja a velocidade programada.

Para o caso específico do manipulador ASEA, não há possibilidade de criar um padrão que atenda a norma ISO devido ao fato do sistema já possuir as relações de engrenagens e resolução de sensores de realimentação pré-estabelecidos. Além disso, este manipulador não tem uma relação direta entre a taxa de pulsos e a velocidade em uma direção cartesiana, pois os movimentos são de junta e não nas direções X, Y e Z, como ocorre nas máquinas cartesianas.

O cálculo da velocidade linear é realizado a partir do cálculo do módulo da cinemática inversa de atuadores entre dois pontos consecutivos discretizados. O resultado da operação deve ser dividido pelo “deltat”, pré-estipulado para cada intervalo discretizado (ANEXO A.1 – linha 8). Desta forma, se a informação de entrada for a velocidade em milímetros/minuto, a cinemática inversa realiza os cálculos informando a quantidade de pulsos necessários para realizar o deslocamento desejado e o “deltat” informa o tempo que o robô terá para fazer cada percurso, como pode ser visto no Anexo A.1, linha de comando 35. Como previsto, o valor de *Feedrate* é diferente da velocidade que o TCP do robô desloca, pois a cinemática inversa leva em consideração em seus cálculos a resolução do *encoder*, a relação das engrenagens harmônicas, além de relações entre os ângulos no movimento de junta a junta. Desta forma, a velocidade linear máxima pode ser facilmente calculada através do programa gerado no Anexo A.1. A velocidade de deslocamento do TCP do robô é de 10mm/s. Nesta velocidade, a frequência de geração de pulsos é de 3.2kHz para o eixo X ou eixo 1, que é o pior caso, ou seja, a maior frequência. A frequência máxima de operação do *driver* é de 20kHz. Uma margem de segurança de 12.5% foi adotada a fim de evitar eventuais falhas do sistema. Assim, a frequência máxima de operação passou a ser de 17.5kHz. A velocidade máxima pode ser calculada dividindo-se 17.5kHz por 3.2kHz e multiplicando o resultado por 10mm/s. O resultado é 55mm/s. Em resultados empíricos, velocidades de 65mm/s foram atingidas, pois o *driver* trabalha em

frequências que chegam a 21.35kHz antes de entrar em falha. Cabe salientar que para cada posição do TCP do robô no plano cartesiano, o valor desta velocidade máxima será diferente, pois em uma posição diferente do Anexo A.1, a taxa de geração de pulsos também será diferente. Isto ocorre devido ao fato dos valores do Espaço de Juntas ser diferente para cada valor de posição no Espaço Cartesiano.

Durante a realização dos movimentos de aproximação e afastamento do robô até coordenadas pré-estabelecidas, deseja-se que a velocidade seja a maior possível, isto é, os *drivers* operem no limite máximo de frequência (20kHz). Para a configuração do computador utilizado nos experimentos, o Mach 3 possui uma relação empírica que correlaciona Kernel, *Feedrate* e taxa de emissão de pulsos, como pode ser verificado na Equação 4.1.

$$K = \frac{Freq}{F}.1504000 \quad (4.1)$$

Onde:

K = frequência de operação do Kernel;

Freq = taxa de emissão de pulsos pelo Mach 3;

F = *Feedrate*.

Desta forma, pode-se estabelecer um *Feedrate* máximo, onde ocorrerá o deslocamento do TCP da ferramenta com a maior velocidade possível sem que os *drivers* entrem em falha. Para um *Feedrate* de 1.290.000 (F1290000), ocorre uma taxa de envio de pulsos de 21.5 kHz para os cinco eixos simultaneamente. A condição mais crítica do movimento de junta é quando a tocha está avançando e o eixo 4 ou A (arfagem da tocha), necessita de 262.144 pulsos para girar 90° para baixo em 15 segundos (ANEXO A.1). Nesta situação, o *driver* está operando a 13.3kHz, com uma margem de 45% de tolerância antes de entrar em falha. Com este movimento, velocidades de 66mm/s são atingidas (movimento de junta a junta).

4.5 Geração de trajetórias com os *softwares* selecionados

O objetivo do *software* gerador da cinemática é permitir que o operador do manipulador robótico programe a posição inicial e a posição final da ferramenta e o tipo de movimento desejado (linear ou circular), juntamente com os parâmetros de velocidade e aceleração. O Anexo A.1 apresenta um programa gerado em MatLab para o experimento em questão. Nesse caso, o objetivo é realizar um deslocamento em X de 100mm, com velocidade de 10mm/s. Para tornar o

programa mais inteligível ao leitor, a justificativa para a utilização de cada linha de comando do código fonte é dada na sequência abaixo:

linha 1: na primeira linha do programa, há uma matriz identidade para informar as dimensões da tocha e stick-out. A matriz de acoplamento da tocha deve ser identidade para que não haja nenhuma interferência na orientação do TCP do robô, ocorrendo somente um incremento de coordenadas devido as dimensões da tocha;

linha 6: nesta linha do código fonte, há a configuração da velocidade a ser desenvolvida pelo TCP do robô. Esta velocidade pode ser variada de 0mm/s até 55mm/s. Velocidades acima de 55mm/s não podem ser atingidas devido a limitações de *hardware*, como foi visto na seção anterior;

linha 7: nesta linha, será programado o valor do passo para que haja a discretização do sinal. Quanto menor for o valor do passo, maior será a discretização, e menor será o erro na trajetória do robô (o movimento fica mais contínuo e suave). Com discretizações de 10mm, ocorre uma movimentação satisfatória do TCP da ferramenta;

linha 8: nesta linha, o “deltat” representa o intervalo de tempo necessário para que o TCP movimente entre dois pontos discretizados subsequentes (será utilizado na linha 35 para cálculo da velocidade em cada intervalo discretizado);

linha 10: na linha 10, é programado o nome do arquivo a ser gerado com o programa;

linha 13: nesta linha, há a programação da coordenada inicial do TCP do robô, juntamente com a orientação da ferramenta, como pode ser visto na Equação 4.2.

$$T^0T_{ini} = \begin{bmatrix} 1 & 0 & 0 & -50 \\ 0 & -1 & 0 & 650 \\ 0 & 0 & -1 & 750 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

Pelo código fonte programado, pode-se ver que se deseja que a ferramenta fique orientada para baixo e que o TCP do robô desloque da posição $X = -50$ até $X = 50$, mantendo Y e Z fixos ($Y = 650$ e $Z = 750$).

Linhas 15 e 16: nestas linhas, os códigos fonte das cinemáticas inversas são chamados. A cinemática inversa de atuadores informará o número de voltas que o motor terá que movimentar para ferramenta chegar até o destino programado. Consequentemente, foi necessário fazer uma

relação entre uma volta do atuador e o número de pulsos para que o mesmo complete esta volta. Para uma resolução do encoder de 8.192 pulsos por revolução, basta multiplicar o valor de “ $A = \text{cininvatu}(\text{teta})$ ” por 8.192 e dividir por 2π ;

linhas 19 a 23: nas linhas 19 a 23, é gerado o código G (ANEXO A.1), onde as variáveis X, Y, Z, A e B representam as juntas 1, 2, 3, 4 e 5, respectivamente. O movimento de aproximação da tocha é do tipo junta a junta. Conseqüentemente, ocorre com velocidade máxima (*Feedrate* máximo – F1.290.000);

linha 27: nesta linha, é informado o deslocamento de 100mm na direção X com passo de 10mm;

linhas 29 e 30: nestas linhas, são chamadas as cinemáticas inversas, gerando a quantidade de pulsos necessária para o movimento de 100mm do TCP;

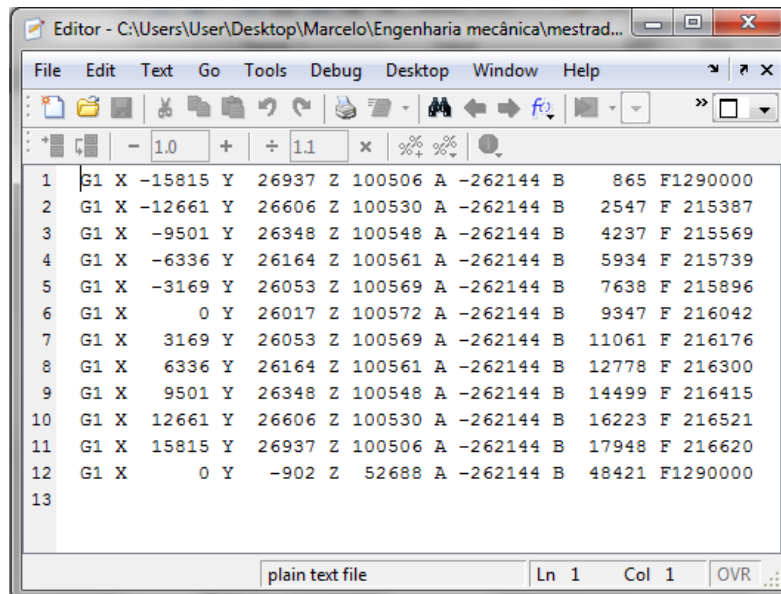
linha 38: nesta linha, é configurado o valor de uma constante K que é calculada de acordo com o valor de Kernel e da configuração do computador que está sendo utilizado. Pelo código G gerado na Figura 4.13, pode-se ver que a diferença de pulsos entre duas linhas consecutivas tem o valor médio de 5.120 pulsos gerados. Para o programa Mach 3 gerar pulsos a uma frequência de 5.120Hz (velocidade = 10mm/s), um valor de *Feedrate* em torno de 215.000 deve ser gerado, como pode ser verificado pela Equação 4.2. Desta forma, pode se calcular o valor de K, dividindo o valor de *Feedrate* pelo valor da taxa de pulsos gerada (215.000/5120). Realizando os cálculos, um valor de $K = 42$ é encontrado. Esta constante não pode ser utilizada diretamente no programa pois a linha 39 não gera o valor da diferença de pulsos entre duas linhas consecutivas e sim o valor do módulo. Assim, este valor de K deverá ser ajustado para atender a uma frequência de 5.120Hz;

linha 39: esta linha possui a fórmula para cálculo da velocidade do TCP do robô. A velocidade é calculada através do módulo da diferença entre a quantidade de pulsos de dois pontos consecutivos dividido pelo “deltat”. Desta forma, pode-se calcular a taxa de pulsos em cada segmento discretizado.

O vetor posição do robô em relação ao referencial inercial é representado pelo somatório de cinco vetores representados pelas juntas X, Y, Z, A e B. Deste modo, ao calcular o módulo do vetor diferença entre duas posições consecutivas, não será obtido o valor real de pulsos que devem ser emitidos e sim o seu módulo. Por exemplo, na Figura 4.13, o módulo entre duas linhas consecutivas tem seu valor médio de 3.600 pulsos. Na prática, se um valor de 3.600Hz for aplicado ao robô, ele terá uma velocidade inferior a 10mm/s (7,7mm/s). Conseqüentemente, um

valor de correção deve ser aplicado a constante K. Este fator de correção é calculado a partir da divisão do valor real de pulsos dividido pelo módulo, isto é $5120/3600$. O valor de K encontrado é 60;

linhas 51 a 55: nestas linhas, é programado um ponto de descanso com velocidade máxima (Feedrate máximo – F1.290.000).



```
1 G1 X -15815 Y 26937 Z 100506 A -262144 B 865 F1290000
2 G1 X -12661 Y 26606 Z 100530 A -262144 B 2547 F 215387
3 G1 X -9501 Y 26348 Z 100548 A -262144 B 4237 F 215569
4 G1 X -6336 Y 26164 Z 100561 A -262144 B 5934 F 215739
5 G1 X -3169 Y 26053 Z 100569 A -262144 B 7638 F 215896
6 G1 X 0 Y 26017 Z 100572 A -262144 B 9347 F 216042
7 G1 X 3169 Y 26053 Z 100569 A -262144 B 11061 F 216176
8 G1 X 6336 Y 26164 Z 100561 A -262144 B 12778 F 216300
9 G1 X 9501 Y 26348 Z 100548 A -262144 B 14499 F 216415
10 G1 X 12661 Y 26606 Z 100530 A -262144 B 16223 F 216521
11 G1 X 15815 Y 26937 Z 100506 A -262144 B 17948 F 216620
12 G1 X 0 Y -902 Z 52688 A -262144 B 48421 F1290000
13
```

FIGURA 4.13. Código G gerado através das cinemáticas inversas e do programa de geração de trajetória.

Após gerar o código G em arquivo do tipo “.txt” (arquivo texto), o código é carregado no programa Mach 3. Antes de iniciar o código, o robô deve sempre estar em sua posição zero de referência, pois é a partir deste ponto que as matrizes de cinemática inversa são criadas.

Para o robô, uma aceleração ideal seria infinita. Porém, para a aceleração do TCP do robô ser infinita, os atuadores teriam que fornecer uma potência infinita para atingir essa aceleração, o que não pode ser implementado na prática. Desta forma, valores de aceleração e desaceleração foram gerados no Mach 3 e testados empiricamente, de acordo com a Tabela 4.2.

TABELA 4.2 - Resposta do manipulador ASEA para diferentes valores de aceleração.

Aceleração do TCP da ferramenta (mm/s^2)	Resposta do manipulador ASEA
9.26	Lenta.
17.36	Excelente.
25.25	<i>Drivers</i> entram em falha, devido a frequência excessiva.

Testes de posicionamento, velocidade e aceleração com carga dinâmica

Após a programação em MatLab, carregar o programa no Mach 3 e ajustar a rampa de aceleração, foi colocado no TCP uma carga de 6kg, a fim de simular a pior condição de operação do robô. Para verificar a trajetória do robô, foram marcados pontos de referência em uma mesa (ponto inicial (P1) e final (P2) – FIG. 4.14) e o tempo de deslocamento entre estes dois pontos foi mensurado com o auxílio de um cronômetro.

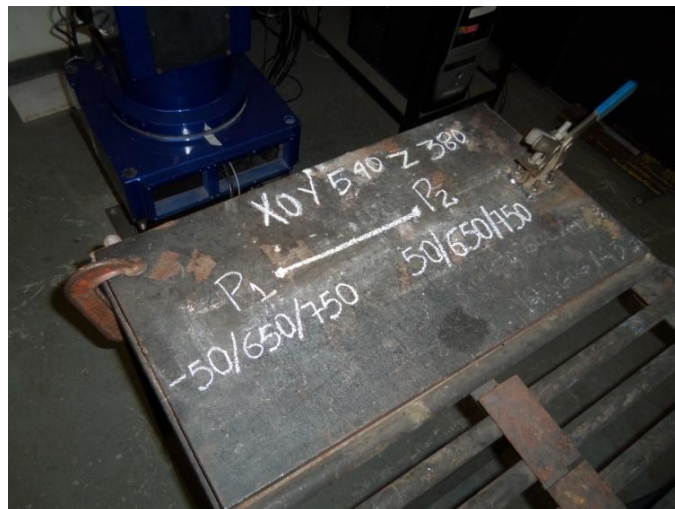


FIGURA 4.14 - Marcação dos pontos de referência para verificação de posicionamento.

Foi observado que o manipulador ASEA obteve boa precisão de posicionamento, pois o robô deslocava-se nos pontos programados sem gerar erros perceptíveis com a utilização de um paquímetro (precisão de 0.05mm) e a velocidade ficou um pouco abaixo do valor programado (em torno de 9.6mm/s), o que pode ser explicado pelo fato da aceleração inicial e desaceleração final não serem infinitas.

4.6 Arrefecimento do gabinete de controle

Através do uso de um termômetro a laser, foram medidas as temperaturas dos *drivers* que variaram de acordo com a carga do robô. Para operação em vazio (sem carga), a temperatura mensurada foi de 47°C. Quando o robô operou em carga máxima por um período de duas horas, a temperatura verificada no termômetro foi de 48°C. De acordo com o fabricante, este equipamento eletrônico pode operar em temperaturas de até 70°C. Portanto, não foi proposto nenhum dispositivo para arrefecimento do gabinete de controle.

4.7 Execução das soldas através do processo *FCAW*

Para validar o processo de remanufatura, o manipulador ASEA foi aplicado em operações de soldagem. A Figura 4.15 apresenta o primeiro experimento, onde o robô foi utilizado para executar uma solda sobre uma chapa com dimensões 6x300, de aço 1020. Durante a execução da solda, optou-se por uma tensão de soldagem de 23V e velocidade de deslocamento da tocha de 20mm/s, *stick-out* de 18mm, tempo de ativação da tocha de 500ms e discretização dos pontos de soldagem a cada 10mm.



FIGURA 4.15 - Cordão de solda obtido no Experimento 1.

O resultado obtido foi uma solda satisfatória, mas alguns parâmetros de soldagem tiveram que ser modificados com o intuito de aumentar o aporte térmico do processo. Os parâmetros *stick-out* e discretização foram mantidos os mesmos em todos os experimentos. Foi adotado um *stick-out* de 18mm por ser um parâmetro muito utilizado em processos de soldagem *FCAW*. Com a discretização de 10mm, uma nova relação de *Feedrate* é recalculada a cada 10mm, juntamente com um novo ponto de referência, o que forneceu cordões de solda homogêneos. Este valor de 10mm foi arbitrado empiricamente, sendo que valores menores de discretização não apresentaram diferenças visuais nos aspectos do cordão de solda.

Nos experimentos subsequentes, os parâmetros foram modificados até ser obtida uma solda com excelente qualidade e aporte térmico elevado (Experimento 5 – FIG. 4.16). No Experimento 5, os parâmetros de soldagem utilizados foram:

- corrente de soldagem – 336A;
- tensão de soldagem – 33V;
- velocidade de deslocamento da tocha – 9,6mm/s;
- discretização do deslocamento – 10mm;
- stick-out – 18mm;
- tempo de ativação da tocha – 100ms;
- comprimento do cordão – 150mm.



FIGURA 4.16 - Cordão de solda obtido no Experimento 5.

Os parâmetros utilizados no Experimento 5 foram os que apresentaram melhores características do cordão de solda. Conseqüentemente, esses parâmetros foram utilizados para enchimento de um chanfro em uma chapa de aço API5LX80 de dimensões 19X250. O chanfro é em meio V e o ângulo é de 45°. O passe de raiz foi realizado manualmente através do processo de soldagem TIG.

Foram necessários sete passes de solda para o enchimento completo do chanfro. A cada passe, o programa em código G era gerado novamente, adicionando-se 2mm na direção Z e variando Y de forma que o cordão ficasse no ponto desejado. Não houve a necessidade de reorientação da ferramenta após cada passe.

A Figura 4.17 (a) apresenta os cordões de solda obtidos no Experimento 8. Na Figura 4.17 (b), pode-se observar uma macrografia do cordão de solda, onde pode ser observada uma inclusão de escória na região central da chapa (seta vermelha). O defeito é devido às imperfeições do processo de soldagem que não estão relacionados com o movimento gerado pelo manipulador ASEA. A inclusão de escória é, geralmente, causada pela limpeza ineficiente entre passes.

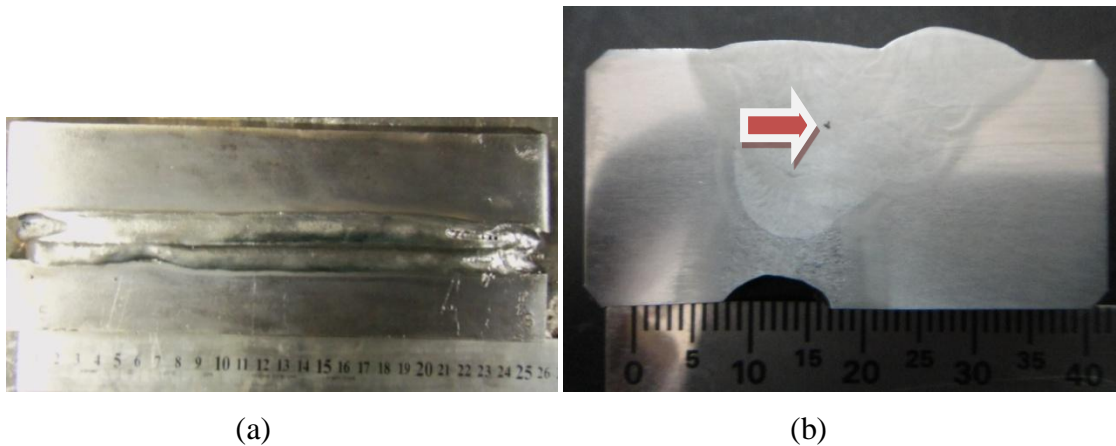


FIGURA 4.17 - (a) Vista de cima da solda em chanfro obtida no Experimento 8 e (b) macrografia, nital 5%.

4.8 Programa em MatLab para movimentação em qualquer orientação no espaço de trabalho

O passo seguinte foi criar um programa em que o TCP da ferramenta possa movimentar em qualquer orientação (ANEXO A.4), desde que este não ultrapasse o espaço de trabalho do robô.

Para atender ao objetivo de criar um programa para deslocamento linear com qualquer orientação, foi necessária a criação de dois tipos de funções que correspondem aos movimentos lineares (ANEXO A.5) e movimentos de junta a junta (ANEXO A.6). Nos movimentos lineares, os parâmetros da função são ponto inicial, ponto final, dimensões da tocha, velocidade da tocha, passo e arquivo para geração do código G. Já para o movimento de junta a junta é fornecido apenas o ponto de descanso, pois é necessário ir ao ponto de descanso o mais rápido possível independentemente da trajetória. Os códigos gerados nos Anexos A.4, A.5 e A.6 possuem linhas de comando similares aos comandos utilizados no Anexo A.1. Desta forma, serão justificadas apenas as linhas de comando que são peculiares a cada programa, conforme descrito abaixo:

- Programa para geração do código G para deslocamento linear em qualquer orientação (ANEXO A.4):
 - linhas 12,13 e 14: nestas linhas, os pontos iniciais e finais de cada movimento linear são programados;
 - linhas 18 e 22: nestas linhas, as funções com os movimentos lineares são chamadas;
 - linha 26: na linha 26, a função com o movimento de junta a junta é chamada.

- Programa para geração da função ASEA movimento linear – ASEA_moveL (ANEXO A.5):
 - linhas 18 e 19: nestas linhas, os pontos iniciais e finais de cada movimento linear são declarados, como os elementos de translação, dentro da matriz transformação homogênea;
 - linha 23: nesta linha, é calculado o comprimento do deslocamento a ser percorrido;
 - linha 24: na linha 24, é gerado um vetor unitário “dirP” na direção de deslocamento da tocha;
 - linha 28: nesta linha, é calculado “P” que indica o ponto final. Este cálculo é feito através da soma vetorial entre o ponto inicial e o vetor gerado pela multiplicação de “dirP” pelo comprimento.
- Programa para geração da função ASEA movimento junta a junta – ASEA_moveJ (ANEXO A.6):
 - linha 2: nesta linha, uma função com o movimento de junta a junta é chamada. Esta função é similar quando se programa o ponto de descanso no Anexo A.1.

Para validar o programa, foi realizado um deslocamento de 50mm na direção X e, na sequência, um deslocamento de 40mm na direção Y, com velocidade de 10mm/s. O resultado foi satisfatório, pois o robô realizou a trajetória com precisão.

Com a utilização do programa ASEA_moveL, há a possibilidade de realizar trajetórias mais complexas, desde que esses movimentos possam ser subdivididos em deslocamentos lineares.

A Figura 4.18 apresenta a Célula Robótica constituída pelo manipulador ASEA, após o processo de *overhauling*. Com essa Célula Robótica, foram simulados 2.500 ciclos de trajetória de um chanfro de 100mm de comprimento durante 12 horas. Os resultados foram satisfatórios, pois o manipulador apresentou boa repetibilidade.



FIGURA 4.18 – Célula Robótica do LRSS constituída pelo manipulador ASEA, após o processo de *overhauling*.

5. CONSIDERAÇÕES FINAIS

Utilizando as técnicas descritas nesta dissertação, pode ser desenvolvida uma plataforma padrão para a remanufatura de manipuladores robóticos industriais. Esta plataforma passa a ser uma alternativa viável para empresas de pequeno e médio porte, tendo em vista que o custo final da remanufatura é menor do que o de um equipamento novo equivalente.

Na realização da remanufatura, existem dois conceitos importantes que devem ser avaliados antes que o processo seja iniciado. Em uma linha de pesquisa de remanufatura, o *retrofitting* busca ao máximo o reaproveitamento de sistemas do manipulador robótico que estão em operação, fazendo adaptações e atualizações a novas tecnologias. Já na metodologia criada neste trabalho, denominada *overhauling*, ocorre somente o reaproveitamento da estrutura mecânica em manipuladores robóticos que possuem idade avançada. Desta forma, o processo de *overhauling* foi adotado para a realização da remanufatura do manipulador ASEA, pois o robô possui idade avançada, e seus componentes eletrônicos e eletromecânicos não possuem histórico de utilização, sendo uma tarefa complexa levantar um parecer técnico favorável ou desfavorável a sua reutilização. Apesar de o *overhauling* aumentar em 35% o custo da remanufatura quando comparado ao *retrofitting* do manipulador ASEA, este processo de reciclagem é mais simplificado, com maior confiabilidade e redução do tempo gasto. Mesmo com o aumento do custo de remanufatura, o *overhauling* representa apenas 14% do valor de um robô novo com características similares com preço de mercado de U\$17.500,00.

Outro acréscimo importante que o processo de remanufatura do manipulador ASEA trouxe foi que, durante a seleção de *hardware* e *software*, optou-se por uma arquitetura aberta na construção do gabinete de controle e uma linguagem de programação não proprietária, onde a geração de cinemática pelo MatLab não é acoplada a um sistema operacional dedicado. Desta forma, o manipulador ASEA torna-se uma ferramenta atrativa ao meio acadêmico, pois sua arquitetura e estrutura de acionamento podem ser facilmente aplicadas a outros processos. A linguagem de programação MatLab facilita a realização de modificações ao sistema bem como a implementação de novas tarefas e o gabinete de controle é intercambiável com outras plantas.

6. SUGESTÕES PARA TRABALHOS FUTUROS

Dentre as implementações e modificações que podem ser realizadas, podem ser citadas as seguintes:

- Implementação de *softwares* para a realização de tarefas de alto nível de programação. Como exemplos, podem ser citados a criação de *softwares* para a identificação de chanfros em peças a serem soldadas, através de visão computacional.
- Utilização da arquitetura proposta em outras áreas como a bioengenharia. Através da arquitetura de *hardware* e *software* criada, podem ser construídas próteses para os membros superiores e inferiores.
- Utilização do manipulador ASEA para outras operações além da soldagem. Através da troca da ferramenta presente no TCP, o manipulador robótico poderá participar de outras operações de manufatura como usinagem, paletização, pintura, inspeção e aplicações que exigem precisão, repetibilidade e são maléficas à saúde humana.
- Substituição dos *softwares* MatLab e Mach 3 por microcontroladores. Os *softwares* utilizados neste experimento são proprietários e necessitam de licenças de alto custo para operação. Conseqüentemente, para produção em larga escala, estes *softwares* podem ser substituídos por microcontroladores.
- Troca dos *encoders* atuais por sensores de menor resolução. Desta forma, a velocidade máxima de 55mm/s pode chegar até valores próximos de 1.100mm/s (*encoder* com resolução de 400 pulsos por revolução, ao invés de 8.192);
- Desenvolvimento de técnicas quantitativas para sintonia do controlador PID. Poderiam ser levantadas técnicas para a modelagem matemática da função de transferência do sistema. Assim, os parâmetros das ações de controle proporcional, integral e derivativa poderiam ser calculados ao invés de serem ajustadas empiricamente;
- Desenvolvimento de um módulo de comunicação, no qual os dados são enviados ao gabinete de controle através de um protocolo TCP/IP, possibilitando o controle do manipulador ASEA remotamente via internet.

7. REFERÊNCIAS BIBLIOGRÁFICAS

BOMFIM, M.H.S. **Projeto de um gabinete de controle com interface USB para robôs de até seis graus de liberdade**. Trabalho de Graduação em Engenharia Mecânica. UFMG. Belo Horizonte, 2009, 46 p.

BOMFIM, M.H.S.; BRACARENSE, A.Q.; LIMA II, E.J.; GONTIJO, R.A. ***Overhauling of an ASEA robot IRB6 with open architecture***. 11th IEEE INTERNATIONAL CONFERENCE ON TRUST, SECURITY AND PRIVACY IN COMPUTING AND COMMUNICATIONS (TrustCom-2012). Liverpool, UK, 2012, pp. 482-489.

BRACARENSE, A.Q.; CORDEIRO, J.F.; KIENITZ, K.M.; BOMFIM, M.H.S.; Souza, A.J.A.; FILHO, F.A.R.; LIMA II, E.J. ***Development of control cabinet for industrial robots up to six degrees of freedom***. 20th INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING. Gramado, BR, 2009, pp. 427-432.

BRACARENSE, A.Q.; MARQUES, P. V.; MODENESI, P. J. **Soldagem Fundamentos e Tecnologia**. 2^a edição. Belo Horizonte: Editora UFMG, 2007, pp. 255-261.

BRACARENSE, A.Q.; VASCONCELOS, D.C.M.; TORRES, G.C.F.; HENRIQUES, R.V.B. ***Retrofitting do robô ASEA IRB6***. 2^o CONGRESSO BRASILEIRO DE ENGENHARIA DE FABRICAÇÃO. Uberlândia, BR, 2003.

CHAPMAN, S. J. **Programação em MatLab para Engenheiros**. 2^a edição. São Paulo: Editora CENGAGE, 2011, pp. 38-251.

CRAIG, J. J. ***Introduction to Robotics Mechanics and Control***. 2^a edição. São Paulo: Editora Addison-Wesley Publishing Company, 1989, pp. 1-144.

FERRARI, C.D. **Implantação de um sistema de Comando Numérico Computadorizado (CNC) por *Retrofitting* de um torno mecânico universal**. Trabalho de graduação em Engenharia de Controle e Automação. FATEC. Catanduva, 2011, 16p.

FITZGERALD, A.E.; KINGSLEY JUNIOR, C.; UMANS, S.D. **Máquinas Elétricas**. 6^a edição. Porto Alegre: Editora Bookman, 2006, pp. 239-389.

GLEIZER, G.; DE LIMA E SILVA, A.; ORENSTEIN, L.P.; LIZARRALDE, F.; LEITE. **Atualização de *hardware* e *software* de um robô industrial**. 18^o CONGRESSO BRASILEIRO DE AUTOMÁTICA. Bonito, BR, 2010, pp. 4403-4410.

GROOVER, M.P. **Robótica: tecnologia e programação**. 2ª edição. São Paulo: Editora McGraw-Hill, 1989, pp. 10-89.

KENJO, T. *Electric Motors and Their Controls*. 1ª edição. São Paulo: Editora Oxford Science Publications, 1991, pp. 1-143.

LIMA II, E.J. **Soldagem Robotizada com Eletrodo Revestido**. Qualificação de doutorado. Programa de Pós-Graduação em Engenharia Mecânica da UFMG. Belo Horizonte, 2005, 89 p.

LIMA II, E.J.; SOUZA, M.B.; ALMEIDA, A.G.S.; RIBEIRO A.S. **Metodologia para implementação de Retrofitting de controladores de equipamentos de automação de processos**. 8º CONGRESSO IBEROAMERICANO DE ENGENHARIA MECÂNICA. Cusco, PE, 2007.

MAIA, A.A.T. **Metodologia de desenvolvimento de um algoritmo para controle simultâneo da capacidade de refrigeração e do grau de superaquecimento de um sistema de refrigeração**. Tese de doutorado. Programa de Pós-Graduação em Engenharia Mecânica da UFMG. Belo Horizonte, 2005, pp. 116-117.

MALVINO, A. P. **Eletrônica**. 1ª edição. São Paulo: Editora McGraw-Hill, 1987, pp. 22-174.

MODENESI, P. J. & MARQUES, P. V. Apostila: Soldagem I – **Introdução aos Processos de Soldagem**. 2006.

OGATA, K. **Engenharia de Controle Moderno**. 4ª edição. São Paulo: Editora Pearson Prentice Hall, 2003, p.788.

PENA, J.C.O.; RAMALHO FILHO, F. A.; CORDEIRO, J. F.; KIENITZ, K. SILVA, E. B.; LIMA II, E. J.; BRACARENSE, A. Q. **Development of a Low Cost CNC Plasma Cutting Machine**. 21st BRAZILIAN CONGRESS OF MECHANICAL ENGINEERING (COBEM 2011). Natal, RN, BR, 2011, pp. 1-10.

PINTO, D. P. **Modelagem da forma geométrica de cordões de solda utilizando ajuste de curvas para simulação computacional**. Dissertação de mestrado. Programa de Pós-Graduação em Engenharia Mecânica da UFMG. Belo Horizonte, 2011, pp. 23-25.

ROMANO, V. F. **Robótica Industrial**. 1ª Edição. São Paulo: Editora Edgard Blucher, 2002, pp. 110-240.

ROSÁRIO, J. M. **Princípios de Mecatrônica**. 1ª Edição. São Paulo: Editora Pearson Prentice Hall, 2005, pp. 141-281.

SILVA, A.M.; PINHEIRO, M. S. F.; FREITAS, N.E. **Guia para normalização de trabalhos técnico – científicos: Projetos de pesquisa, monografias, dissertações, teses**. 2ª Edição. Uberlândia: Editora da Universidade Federal de Uberlândia, 2002, 159 p.

SILVA, R.F. **Controle de instrumentos digitais de medição por meio de interface USB com uso de padrão SCPI**. Dissertação de mestrado. Programa de Pós-Graduação em Engenharia Elétrica da UFMG. Belo Horizonte, 2003, 152p.

SZKODNY, T. *Forward and inverse kinematics of irb-6 manipulator. Mechanism and Machine Theory*, 1995, volume 30, p.1039-1056.

VORPÉRIAN, V. *Simplify PWM Converter Analysis Using a PWM Switch Model*. 1ª Edição. São Paulo: Editora PCIM, March 1990, pp. 10-15.

7.1 Sítios visitados na internet

(CNC4, 2012). Disponível em: <<http://www.CNC4.com/>>. Acesso em: 18/09/2012.

(GECKODRIVE, 2012). Disponível em: <<http://www.geckodrive.com/>>. Acesso em: 03/05/2012.

(KUKA, 2012). Disponível em: <<http://www.KUKA.com/>>. Acesso em: 07/10/2012.

(MACHSUPPORT, 2012). Disponível em: <<http://www.machsupport.com/>>. Acesso em: 18/06/2012.

(TOSHIBA, 2012). Disponível em: <<http://www.toshiba.com/>>. Acesso em: 18/07/2012.

(TOYOTA, 2012). Disponível em: <<http://www.toyota.com/>>. Acesso em: 06/10/2012.

(USDIGITAL, 2012). Disponível em: <<http://www.USDigital.com/>>. Acesso em: 20/05/2012.

(WEG, 2012). Disponível em: <<http://www.WEG.com.br/>>. Acesso em: 08/10/2012.

ANEXO A

A.1 Programa para geração do código G para deslocamento linear em x (solda de 100mm com velocidade de 10mm/s).

```
T5T=eye(4); %Matriz identidade
vel=10; %mm/min - configurável - velocidade do TCP da ferramenta
passo=10; %mm - configurável - Discretização
deltat=abs(passo/vel); %Intervalo de tempo
fid = fopen('velocidade.txt','w');%Geração de um arquivo txt
%determinacao do ponto inicial
T0T_ini=[1 0 0 -50;0 -1 0 650;0 0 -1 750;0 0 0 1]; %Matriz transformação,
mediu-se em relação a um referencial x,y e z da base
teta=cininv(T0T_ini,T5T);%Está chamando o programa cinemática inversa
A=cininvatu(teta)*2048*4/(2*pi); %Geração de pulsos - (2048*4/2*pi) = pulsos
necessários para um rotação do motor
fprintf(fid,'G1 X %6.0f',A(1)); %Geração do código G para os eixos X,Y,Z,A e
B
fprintf(fid,' Y %6.0f',A(2));
fprintf(fid,' Z %6.0f',A(3));
fprintf(fid,' A %6.0f',A(4));
fprintf(fid,' B %6.0f F1290000\n',A(5));% Feedrate necessário para aumentar
taxa depulsos de acordo com Kernel selecionado
Aant=A;

for x=-50+passo:passo:50,% Deslocamento de 200mm em x
    T0T=[1 0 0 x;0 -1 0 650;0 0 -1 750;0 0 0 1];
    teta=cininv(T0T,T5T); %Está chamando o programa cinemática inversa
    A=cininvatu(teta)*2048*4/(2*pi);%Geração de pulsos - (2048*4/2*pi) =
pulsos necessários para um rotação do motor

    fprintf(fid,'G1 X %6.0f',A(1)); %Geração do código G para os eixos
X,Y,Z,A e B
    fprintf(fid,' Y %6.0f',A(2));
    fprintf(fid,' Z %6.0f',A(3));
    fprintf(fid,' A %6.0f',A(4));
    fprintf(fid,' B %6.0f',A(5));
    K=60;
    feedrate=sqrt((A-Aant)*(A-Aant)')/deltat*K;%Cálculo da velocidade
    Aant=A;
    fprintf(fid,' F %6.0f\n',feedrate);% Feedrate necessário para aumentar
taxa de pulsos de acordo com Kernel selecionado
end

%determinacao do ponto de descanso
T0T_desc=[1 0 0 0;0 -1 0 650;0 0 -1 900;0 0 0 1]; %Arbitrário, posição de
descanso
teta=cininv(T0T_desc,T5T);
A=cininvatu(teta)*2048*4/(2*pi); %Geração de pulsos - (2048*4/2*pi) = pulsos
necessários para um rotação do motor
fprintf(fid,'G1 X %6.0f',A(1));
fprintf(fid,' Y %6.0f',A(2));
fprintf(fid,' Z %6.0f',A(3));
fprintf(fid,' A %6.0f',A(4));
fprintf(fid,' B %6.0f F1290000\n',A(5));
Aant=A;

fclose(fid);
```

A.2 Programa para geração da cinemática inversa (as deduções matemáticas estão na metodologia).

```
function teta = cininv(T0T,T5T);
%L(2)=0.45;
%L(3)=0.67;
%lam(1)=0.7;
%lam(5)=0.095;
lam(1)=0.7 *1000;
L(2)=0.45 *1000; %Workspace
%L(2)=0.45 *1000; %Tadeuz
L(3)=0.67 *1000;
lam(5)=0.095 *1000;

T05=T0T*inv(T5T);
nx=T05(1,1); ox=T05(1,2); ax=T05(1,3); px=T05(1,4);
ny=T05(2,1); oy=T05(2,2); ay=T05(2,3); py=T05(2,4);
nz=T05(3,1); oz=T05(3,2); az=T05(3,3); pz=T05(3,4);

teta(1)=atan2(-px,py);

w1=-sin(teta(1))*px + cos(teta(1))*py + lam(5)*sin(teta(1))*ax -
lam(5)*cos(teta(1))*ay;
w2=pz-lam(1)-lam(5)*az;          %No artigo esta como w2=p2-...

S3=(w1*w1+w2*w2-(L(2)*L(2)+L(3)*L(3)))/(2.0*L(2)*L(3));
teta(3)=asin(S3);
C3=cos(teta(3));

S2=(w2*L(3)*C3-
w1*(L(3)*S3+L(2)))/((L(3)*L(3)*C3*C3)+(L(3)*S3+L(2))*(L(3)*S3+L(2)));
C2=(w1*L(3)*C3+w2*(L(3)*S3+L(2)))/((L(3)*L(3)*C3*C3)+(L(3)*S3+L(2))*(L(3)*S3+L(2)));
teta(2)=atan2(S2,C2);
S2=sin(teta(2));
C2=cos(teta(2));

lam5S34= sin(teta(1))*S2*px-cos(teta(1))*S2*py+C2*pz-lam(1)*C2-L(2)-L(3)*S3;
lam5C34=-sin(teta(1))*C2*px+cos(teta(1))*C2*py+S2*pz-lam(1)*S2-L(3)*C3;

teta34=atan2(lam5S34,lam5C34);

teta(4)=teta34-(teta(3));

S5=cos(teta(1))*nx+sin(teta(1))*ny;
C5=cos(teta(1))*ox+sin(teta(1))*oy;
teta(5)=atan2(S5,C5);
```

A.3 Programa para geração da cinemática inversa de atuadores (as deduções matemáticas estão na metodologia).

```
function A=cininvatu(teta);
k1=-1/158;
A(1)=teta(1)/k1;

%BA=200; %Preliminares
BA=190; %VERIFICAR
BC=240; %VERIFICAR
BD=140; %VERIFICAR
%DJ=440; %Preliminares
DJ=430; %VERIFICAR
BJ=450;
k2=-2.5/pi;
%k21=33.56/180*pi;
%k22=76.93/180*pi;
%lambda2=sqrt(BC*BC+BD*BD-2*BC*BD*cos(k21+k22))
%lambda2=374.367158; %Preliminares
%lambda2=374.3659361; %Comprimento do atuador linear, para que A2=0 ->
teta2=0
aux=cos(-acos(BA/BC)-acos((BJ^2+BD^2-DJ^2)/(2*BJ*BD))+180/180*pi);
lambda2=sqrt(BC^2+BD^2-2*BC*BD*aux);
aux=teta(2)-acos(BA/BC)-acos((BJ^2+BD^2-DJ^2)/(2*BJ*BD))+(180/180)*pi;
A(2)=(sqrt(BC^2+BD^2-2*BC*BD*cos(aux))-lambda2)/k2;

%FE=200; %Preliminares
FE=190; %VERIFICAR
FG=240; %VERIFICAR
FH=140; %VERIFICAR
k3=-2.5/pi;
%k31=33.56/180*pi;
%lambda3=sqrt(FG*FG+FH*FH-2*FG*FH*cos(k31+90/180*pi))
%lambda3=200.134458; %Preliminares
%lambda3=190.115598; %Comprimento do atuador linear, para que A2=0 e A3=0 ->
teta3=0
lambda3=sqrt(FG^2+FH^2-2*FG*FH*cos(-acos(FE/FG)+90*pi/180));
A(3)=(sqrt(FG^2+FH^2-2*FG*FH*cos(teta(3)+teta(2)-acos(FE/FG)+0.5*pi))-
lambda3)/k3;

k4=-1/128;
A(4)=-(teta(4)+teta(2)+teta(3))/k4;

k5=-1/128;
k6=1.5;
A(5)=(teta(5)/k6+teta(4))/k5;
```


A.4 Programa para geração do código G para deslocamento linear em qualquer orientação.

```
passo=15; %mm - configuravel (discretização)
%Configuração da tocha (L350+Stick-out 20)
T5T=[ 1    0    0    0;
      0    1    0    0;
      0    0    1   370;
      0    0    0    1];

% Geração do arquivo txt
fid = fopen('prognc.txt','w');

%determinacao dos pontos
T0T_1=[1 0 0 0;0 -1 0 740;0 0 -1 650;0 0 0 1];
T0T_2=[1 0 0 0;0 -1 0 727;0 0 -1 840;0 0 0 1];
T0T_3=[1 0 0 0;0 -1 0 785;0 0 -1 900;0 0 0 1];
T0T_4=[1 0 0 0;0 -1 0 785;0 0 -1 400;0 0 0 1];
T0T_5=[1 0 0 0;0 -1 0 765;0 0 -1 520;0 0 0 1];
T0T_1=[1 0 0 0;0 -1 0 740;0 0 -1 650;0 0 0 1];

%Configuração do movimento linear de 1 a 2
%(Ponto1,Ponto2,tocha,velocidade,passo,código G)
ASEA_moveL(T0T_1,T0T_2,T5T,50,passo,fid);

%Configuração do movimento linear de 2 a 3
%(Ponto2,Ponto3,tocha,velocidade,passo,código G)
ASEA_moveL(T0T_2,T0T_3,T5T,50,passo,fid);

%Configuração do movimento linear de 3 a 4
%(Ponto2,Ponto3,tocha,velocidade,passo,código G)
ASEA_moveL(T0T_3,T0T_4,T5T,50,passo,fid);

%Configuração do movimento linear de 4 a 5
%(Ponto2,Ponto3,tocha,velocidade,passo,código G)
ASEA_moveL(T0T_4,T0T_5,T5T,50,passo,fid);

%Configuração do movimento linear de 5 a 1
%(Ponto2,Ponto3,tocha,velocidade,passo,código G)
ASEA_moveL(T0T_5,T0T_1,T5T,25,passo,fid);

%Configuração do movimento de junta
%(Movimento de junta até o ponto de descanso)
ASEA_moveJ(T0T_desc,T5T,fid);

fclose(fid);
```

A.5 Programa para geração da função ASEA movimento linear – ASEA_moveL

```
%Função gerada para programação de movimentos lineares
function ASEA_moveL(T0T_ini,T0T_fin,T5T,vel,passo, fid)

%Cálculo do tempo necessário de movimentação
deltat=passo/vel;

%Deslocamento junta a junta até T0T_ini
teta=cininv(T0T_ini,T5T);
A=cininvatu(teta)*2048*4/(2*pi); %pulsos
fprintf(fid, 'G1 X %6.0f',A(1));
fprintf(fid, ' Y %6.0f',A(2));
fprintf(fid, ' Z %6.0f',A(3));
fprintf(fid, ' A %6.0f',A(4));
fprintf(fid, ' B %6.0f F1290000\n',A(5));
Aant=A;

%Declaração dos pontos iniciais e finais da matriz
Pini=T0T_ini(1:3,4);
Pfin=T0T_fin(1:3,4);

%Cálculo do comprimento a percorrer (Pitágoras)
comprimento=sqrt((Pfin-Pini)'*(Pfin-Pini));
%Criação do vetor unitário direção (dirP)
dirP=(Pfin-Pini)/comprimento;

for s=passo:passo:comprimento;
    %Cálculo da posição (P), para cada valor discretizado
    P=Pini+dirP*s;
    T0T=T0T_ini;
    T0T(1,4)=P(1); T0T(2,4)=P(2); T0T(3,4)=P(3);
    teta=cininv(T0T,T5T);
    A=cininvatu(teta)*2048*4/(2*pi);
    %Geração de pulsos nos 5 eixos
    fprintf(fid, 'G1 X %6.0f',A(1));
    fprintf(fid, ' Y %6.0f',A(2));
    fprintf(fid, ' Z %6.0f',A(3));
    fprintf(fid, ' A %6.0f',A(4));
    fprintf(fid, ' B %6.0f',A(5));
    K=60;
    %Cálculo da taxa de geração de pulsos
    feedrate=sqrt((A-Aant)*(A-Aant)')/deltat*K;
    Aant=A;
    fprintf(fid, ' F %6.0f\n',feedrate);
end
```

A.6 Programa para geração da função ASEA movimento junta a junta – ASEA_moveJ

%Função gerada para programação de movimentos de junta a junta

```
function ASEA_moveJ(T0T_fin,T5T, fid)

teta=cininv(T0T_fin,T5T);
A=cininvatu(teta)*2048*4/(2*pi);
fprintf(fid, 'G1 X %6.0f',A(1));
fprintf(fid, ' Y %6.0f',A(2));
fprintf(fid, ' Z %6.0f',A(3));
fprintf(fid, ' A %6.0f',A(4));
fprintf(fid, ' B %6.0f F1290000\n',A(5));
```