



William Schroeder Cardozo

**Controle de Motores de Passo em Malha Fechada
Aplicado a um Manipulador Robótico**

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Mecânica da PUC-Rio como requisito parcial para obtenção do título de Mestre em Engenharia Mecânica.

Orientador: Prof. Marco Antonio Meggiolaro

Rio de Janeiro
Fevereiro de 2012



William Schroeder Cardozo

**Controle de Motores de Passo em Malha Fechada
Aplicado a um Manipulador Robótico**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Mecânica da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Marco Antonio Meggiolaro

Orientador

Departamento de Engenharia Mecânica – PUC-Rio

Prof. Hans Ingo Weber

Departamento de Engenharia Mecânica – PUC-Rio

Prof. Max Suell Dutra

COPPE - UFRJ

Prof. José Eugênio Leal

Coordenador Setorial do CTC-PUC-Rio

Rio de Janeiro
Fevereiro de 2012

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

William Schroeder Cardozo

Graduou-se em Engenharia Mecânica pela PUC-Rio (Pontifícia Universidade Católica) em 2006. Trabalhou por três anos no desenvolvimento de máquinas operatrizes para retífica de motores antes de começar o mestrado na Área de Mecânica Aplicada na Pós-Graduação da PUC-Rio. Suas áreas de interesse abrangem robótica, controle e modelagem de sistemas e desenvolvimento de máquinas de controle numérico.

Ficha Catalográfica

Cardozo, William Schroeder

Controle de Motores de Passo Aplicado a um Manipulador Robótico / William Schroeder Cardozo; orientador: Marco Antonio Meggiolaro. – Rio de Janeiro: PUC, Departamento de Engenharia Mecânica, 2012.

v., 141 f.: il. ; 29,7 cm

1. Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica.

Incluir referências bibliográficas.

1. Engenharia Mecânica – Teses. 2. Robótica Industrial. 3. Motores de Passo. 4. Controle. I. Meggiolaro, Marco Antonio. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. III. Título.

à minha esposa, Aline, e aos meus pais, Sebastião e Lucélia

Agradecimentos

Ao meu orientador Professor Marco Antonio Meggiolaro pelo estímulo e parceria para a realização deste trabalho.

À PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Aos meus pais, pela educação, atenção e carinho de todas as horas.

Ao meu professor Hans Ingo Weber, pelos vários ensinamentos.

Aos meus colegas da PUC-Rio.

Aos professores que participaram da banca examinadora.

A todos os professores e funcionários do Departamento pelos ensinamentos e pela ajuda.

A todos os amigos e familiares que de uma forma ou de outra me estimularam ou me ajudaram.

Resumo

Cardozo, William Schroeder; Meggiolaro, Marco Antonio. **Controle de Motores de Passo Aplicado a um Manipulador Robótico**. Rio de Janeiro, 2012. 141p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Motores de passo são os motores mais utilizados em aplicações de controle de posicionamento em malha aberta. Entretanto, as limitações desta forma de atuação têm fomentado o desenvolvimento de novas técnicas que incorporem o controle em malha fechada. Motores de passo possuem boa relação entre torque e custo, tornando-os atraentes para aplicações em manipuladores robóticos. Mas as técnicas tradicionais de controle de manipuladores elétricos, que normalmente assumem o uso de motores de corrente contínua, apresentam baixo desempenho quando aplicadas a motores de passo, mesmo com o uso de sensores de posição. A forma mais comum de controle em malha fechada de motores de passo exige um *encoder* diretamente acoplado ao eixo do motor, formando um “sistema colocado”. No entanto, o projeto de muitos motores de passo não permite este acoplamento. Nesses casos, é necessário instalar os *encoders* na estrutura do manipulador, separados dos atuadores, caracterizando um “sistema não-colocado”, que tipicamente apresenta problemas de estabilidade. Este trabalho propõe uma técnica de controle que recebe a realimentação de um *encoder*, não diretamente acoplado ao motor, e gera uma sequência de pulsos para o *driver* do motor de passo. Esse trem de pulsos é calculado de modo a não exigir acelerações excessivas, e assim prevenir a perda de passo do motor. O modelo de um sistema robótico usando este controlador é desenvolvido e simulado em Simulink/MATLAB. Um manipulador robótico de seis graus de liberdade acionado por motores de passo é especialmente projetado e construído para validar a técnica de controle apresentada, controlado por um microcontrolador PIC18F2431. O manipulador desenvolvido é modelado, e sua dinâmica analisada através de simulações. Os experimentos comprovam a eficiência da técnica de controle proposta, resultando em uma precisão absoluta na extremidade do manipulador de $\pm 1,3mm$ e repetibilidade $\pm 0,5mm$.

Palavras-chave

Robótica; Manipuladores Robóticos; Motores de Passo; Controle de Posição.

Abstract

Cardozo, William Schroeder; Meggiolaro, Marco Antonio. **Stepper Motores Control Applied to a Robotic Manipulator**. Rio de Janeiro, 2012. 100p. Msc. Dissertation – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Stepper motors are used in most applications in open loop. However, the limitations of this type of control have encouraged the development of new techniques for closed loop control. Stepper motors have a good relationship between torque and cost, making it attractive for applications in robotic manipulators. But the limitation of traditional control deteriorates the performance of the manipulator. The most common form of closed loop control of stepper motors require an encoder directly coupled to the motor shaft. However, this is not always practical. In some cases, it is necessary to control the position of some system component that can't be precisely known from the position of the motor. This work proposes a control technique that receives feedback from an encoder, not directly coupled to the motor shaft, and generates a sequence of pulses to the stepper motor driver. This pulse train is done so as not to require excessive accelerations, and thus prevent the loss of step. The model of a system using this controller is built using Simulink/MATLAB. A robotic manipulator of six degrees of freedom, using stepper motors, is designed and built to validate the presented control techniques, implemented on a PIC18F2431 microcontroller. The obtained absolute accuracy is $\pm 1,3mm$ and repeatability $\pm 0,5mm$, proving the efficiency of the proposed control technique.

Keywords

Mechanical Engineering; Control; Robotics; Stepper Motor.

Sumário

1	Introdução	17
2	Fundamentos teóricos	21
2.1.	Motores de passo	21
2.2.	Motores de passo em malha aberta	26
2.3.	Motores de passo em malha fechada	27
2.4.	Redutores com folga (<i>backlash</i>)	30
3	Modelagem de motores de passo	32
3.1.	Circuito equivalente	32
3.2.	Torque gerado	33
3.3.	Equação de movimento do motor	34
3.4.	Relação entre dentes, fases e número de passos	35
3.5.	Cálculo do fluxo induzido máximo	36
4	Controle de motores de passo	37
4.1.	Malha de controle	37
4.2.	Base de regras	39
4.3.	Estimativa das constantes do controlador	40
5	Sistema experimental	45
5.1.	Projeto do manipulador	45
5.2.	Projeto do controlador	57
5.3.	EMC (Enhanced Machine Controller)	60
6	Modelagem do manipulador	65
6.1.	Cinemática direta	65
6.2.	Cinemática inversa	69
6.3.	Dinâmica	75
7	Simulações	79

7.1. Modelo de motor de passo	79
7.2. Modelo de <i>driver</i>	81
7.3. Gerador de pulsos	83
7.4. Controlador	84
7.5. Simulação de um motor com controlador	85
7.6. Ajuste do controlador	89
7.7. Simulação do manipulador	92
7.8. Cálculo do fluxo máximo induzido dos motores utilizados	95
7.9. Resultado da simulação do manipulador	97
8 Experimentos	100
8.1. Calibração do sistema	100
8.2. Sistema de medição	103
8.3. Posições medidas	105
8.4. Cálculo do erro	106
8.5. Resultados	107
9 Conclusões e trabalhos futuros	111
10 Referências bibliográficas	113
Apêndice A – Programas dos microcontroladores	117
Apêndice B – Códigos do MATLAB	125
Função para simulação do manipulador	125
Função para cálculo da matriz de inércia	131
Função para cálculo da cinemática inversa	135
Apêndice C – Dados dos motores e redutores	138

Lista de figuras

Figura 1 – Esquema motor VR	21
Figura 2 – Fluxo motor VR	22
Figura 3 – Motor 4 fases	22
Figura 4 – Driver 4 fases	23
Figura 5 – Sequência motor de 4 fases	23
Figura 6 – Esquema motor híbrido	24
Figura 7 – Sequência motor híbrido	25
Figura 8 – Sinais de comando	27
Figura 9 – Malha fechada típica	28
Figura 10 – Torque vrs. posição por fase	28
Figura 11 – Torque vrs. velocidade para vários ângulos de comutação	29
Figura 12 – Foto de um motor com <i>encoder</i>	29
Figura 13 – Folga entre dentes	30
Figura 14 – Ângulo do eixo de entrada vrs. saída com folga	31
Figura 15 – Circuito equivalente	32
Figura 16 – Esquema ímã híbrido	34
Figura 17 – Dentes rotor e estator	35
Figura 18 – Malha de controle proposta	38
Figura 19 – Gráfico de desempenho de um motor de passo	41
Figura 20 – Gráfico erro vrs. tempo de um sistema	42
Figura 21 – Desaceleração constante do motor	42
Figura 22 – Manipulador robótico	46
Figura 23 – Localização dos motores	47
Figura 24 – Dimensões básicas	48
Figura 25 – Vista explodida das três primeiras juntas	49
Figura 26 – Foto dos motores 1 e 2	50
Figura 27 – Foto dos motores 3 e 4	50
Figura 28 – Foto dos motores 5 e 6	51
Figura 29 – Vista explodida dos três últimos elos	51
Figura 30 – Detalhe das polias do punho	52
Figura 31 – Detalhe do punho	53
Figura 32 – Posição do centro de gravidade dos três últimos elos	54

Figura 33 – Posição do centro de gravidade dos quatro últimos elos	54
Figura 34 – Montagem do <i>encoder</i> na junta 1	55
Figura 35 – Montagem do <i>encoder</i> na junta 2	56
Figura 36 – Montagem dos <i>encoders</i> nas junta 3 e 4	56
Figura 37 – Montagem dos <i>encoders</i> nas junta 5 e 6	57
Figura 38 – Registrador QEI	58
Figura 39 – Diagrama elétrico do controlador	59
Figura 40 – Diagrama do <i>software</i> implementado	62
Figura 41 – Foto do PC utilizado	63
Figura 42 – Diagrama geral de conexões	64
Figura 43 – Diagrama dos eixos coordenados	66
Figura 44 – Posição do centro do punho	69
Figura 45 – Duas soluções da junta 3	71
Figura 46 – Ícone do motor de passo do Simulink	79
Figura 47 – Caixa de diálogo do motor de passo do Simulink	80
Figura 48 – Modelo de <i>driver</i> no Simulink	81
Figura 49 – Entrada e saída do sequenciador	82
Figura 50 – Modelo de gerador de pulsos	83
Figura 51 – Modelo de controlador	84
Figura 52 – Subsistema com algoritmo de controle	84
Figura 53 – Modelo de motor, redutor e controlador	86
Figura 54 – Modelo do <i>encoder</i>	87
Figura 55 – Gráficos do motor	87
Figura 56 – Velocidade máxima do motor	88
Figura 57 – Posição do <i>encoder</i>	88
Figura 58 – Sistema demandado de <i>40inc.</i> de amplitude, a <i>0,05Hz</i>	89
Figura 59 – Sistema demandado de <i>40inc.</i> de amplitude, a <i>0,5Hz</i>	89
Figura 60 – Saída com $K_p = 10$ e $K_D = 0,02$	90
Figura 61 – Saída com $K_p = 15$ e $K_D = 0,02$	91
Figura 62 – Saída com $K_p = 13$ e $K_D = 0,02$	91
Figura 63 – Velocidade do motor vrs. demandada	92
Figura 64 – Modelo com motores e manipulador	93
Figura 65 – Esquema da bancada de medidas	95

Figura 66 – Foto da bancada de medidas	96
Figura 67 – Media de tensão no osciloscópio	96
Figura 68 – Erro da posição da extremidade	97
Figura 69 – Movimento do manipulador	98
Figura 70 – Rastro do manipulador	99
Figura 71 – Nivelamento da base	100
Figura 72 – Calibração da junta 2	101
Figura 73 – Calibração da juntas 3 e 4	101
Figura 74 – Calibração das juntas 5 e 6	102
Figura 75 – Calibração da junta 1	102
Figura 76 – Linhas de referência	103
Figura 77 – Media da altura de um ponto	104
Figura 78 – Media orientação da extremidade	104
Figura 79 – Media com relógio comparador	105

Lista de Símbolos

A_i^{i-1}	Matriz de transformação do sistema $i-1$ para o sistema i
a_i	Comprimento da normal comum entre a junta i e a junta $i-1$
c_i	$\cos(\theta_i)$
c_{ij}	$\cos(\theta_i + \theta_j)$
D	Coefficiente de atrito viscoso
d_i	Distância do ponto onde parte a normal comum
E_M	Tensão máxima medida nos terminais em aberto do motor
e_A	F.E.M. (Força Eletro Motriz) na fase A
e_B	F.E.M. (Força Eletro Motriz) na fase B
e_0	Erro no início da desaceleração
f	Frequência de pulsos de passo
G_i	Torque a gravidade no elo i
\bar{g}	Vetor da força gravitacional
H	Henry
I_i	Matriz de inércia do elo i em relação ao centro de massa
${}^j I_i$	Matriz de inércia do elo i usando coordenadas da junta j
I_M	Corrente máxima na fase
i_A	Corrente na fase A
i_B	Corrente na fase B
i_5	Relação de transmissão da junta 5
i_6	Relação de transmissão da junta 6
J	Inércia do motor de passo mais a carga
J_{vj}^i	J -ésima coluna da matriz Jacobiana linear do elo i
J_{oi}	Sub-matriz Jacobiana associada à velocidade angular do elo i
K_P	Constante de ganho proporcional
K_D	Constante de ganho derivativo
L_A	Indutância na fase A
L_B	Indutância na fase B
M_{ij}	Termo ij da matriz de inércia do manipulador
m	Número de fases
m_j	Massa do elo j
N_r	Número de dentes do rotor
N	Velocidade angular em RPM
${}^{j-1} p_{ci}^*$	Vetor posição do CG de i , em relação $j-1$, escrito na base.
p	Número de pares de pólos magnéticos
${}^i p$	Posição do ponto P no sistema i
Q_i	Torque generalizado do elo i
q	Vetor de coordenadas generalizadas das juntas

q_x	Coordenada x da extremidade
q_y	Coordenada y da extremidade
q_z	Coordenada z da extremidade
R_A	Resistência da fase A
R_B	Resistência da fase B
${}^j r_{ci}$	Vetor posição do CG do elo i , usando coordenadas de j
S	Número de passos por revolução
s_i	$sen(\theta_i)$
s_{ij}	$sen(\theta_i + \theta_j)$
T_{dm}	Torque de retenção
T_L	Torque da carga
T_L	Torque da carga
t	Tempo
t_c	Tempo crítico
u	Vetor unitário na direção de x_6
V_{PD}	Velocidade calculada pela lei de controle PD
V_{MIN}	Velocidade mínima que não ultrapassa máxima desaceleração
V_{MAX}	Velocidade máxima que não ultrapassa máxima aceleração
V_C	Velocidade de saída do controlador
V_S	Velocidade de saturação da saída
V_i	Torque devido termos centrífugos e de Coriolis na junta i
v	Vetor unitário na direção de y_6
v_A	Tensão nos terminais da fase A
v_B	Tensão nos terminais da fase B
w	Vetor unitário na direção de z_6
α_i	Ângulo entre o eixo z_{i-1} e o eixo z_i em relação a x_i
δ_i	Ângulo de rotação do eixo do motor i
θ_i	Ângulo entre o eixo x_{i-1} e o eixo x_i em relação a z_{i-1}
θ	Ângulo de rotação do rotor
θ_S	Ângulo de passo em graus
λ	Defasagem da excitação do motor
τ_e	Torque eletromagnético gerado pelo motor
τ_A	Torque gerado pela fase A do motor
τ_B	Torque gerado pela fase B do motor
Ψ_M	Fluxo induzido máximo
Ψ_A	Fluxo induzido na fase A
Ψ_B	Fluxo induzido na fase B
Ω	Ohm
ω	Frequência angular

Lista de Abreviaturas e Siglas

AC	Corrente Alternada
CAD	Computer-Aided Design
CG	Centro de Gravidade
CNC	Controle Numérico Computadorizado
DC	Corrente Contínua
DH	Denavit Hartenberg
EKF	Extended Kalman Filter (Filtro Kalman Estendido)
EMC	Enhanced Machine Controller
EMF	Electromotive Force (Força Eletromotriz)
HMI	Human Machine Interface (Interface Homem-Máquina)
MA	Malha Aberta
MF	Malha Fechada
NIST	National Institute of Standards and Technology
PID	Proporcional Integrativo Derivativo
RL	Circuito Resistor Indutor
SMC	Sliding Mode Control
RPM	Rotações por Minuto
VR	Variable Reluctance (Relutância Variável)
ppr	Pulsos por revolução

1 Introdução

Motores de passo são atuadores eletromagnéticos capazes de converter pulsos digitais na entrada em rotações incrementais do eixo do rotor. São utilizados em aplicações que necessitam de um movimento rotativo preciso. Esses motores estão difundidos em impressoras, discos rígidos de computador, máquinas-ferramenta e robôs. O interesse neste sistema é motivado por sua precisão, reduzida manutenção e baixo custo.

Motores de passo de ímã permanente fornecem mais torque por unidade de peso e melhor relação entre torque e inércia que motores de corrente contínua (DC) [1]. Além disso, como não utilizam escovas, são mais confiáveis e necessitam de menor manutenção [2]. A maior parte dos manipuladores robóticos industriais e máquinas operatrizes utilizam motores de corrente alternada acoplados a um sensor de posição (servo AC). Este motor também possui baixa manutenção, pois não utiliza escovas, mas seu custo é quatro ou cinco vezes maior que o de um motor de passo de mesma capacidade de torque.

Como o motor de passo realiza rotações incrementais precisas, a partir de uma entrada digital de pulsos, a posição do rotor pode ser estimada sem um sensor de posição. Isso permite um controle preciso em malha aberta. A partir da contagem dos pulsos enviados para o sistema, e sua frequência, a posição e velocidade do motor são estimadas [3]. Entretanto, quando o motor é demandado acima de seu limite, a resposta à excitação da entrada diverge do esperado. Isso cria um erro permanente na estimativa de posição do motor. Em aplicações onde a carga varia significativamente, o sistema deve ser ajustado para a pior condição, a fim de garantir que o motor realize todos os passos programados. Ou seja, em malha aberta, a aceleração imposta ao motor deve estar significativamente abaixo da ótima, para garantir confiabilidade da estimativa de posição. Devido à quantidade de torque disponível no eixo ser substancialmente reduzida com o aumento da velocidade, a velocidade demandada ao motor também deve ser reduzida.

As limitações do controle em malha aberta motivaram o desenvolvimento de técnicas de controle em malha fechada para motores de passo. Algumas técnicas são chamadas de *sensorless*, pois se baseiam em estimar a posição do motor a partir da análise da corrente e tensão nos seus enrolamentos. A estimativa através do retorno da força eletromotriz (Back-EMF) do motor é utilizada em algumas técnicas *sensorless* [4]. A principal desvantagem deste método é a baixa precisão em baixas velocidades, quando a força eletromotriz é muito pequena. Outro método se baseia na variação da indutância do motor em função da posição do rotor [5]. Os dois métodos também são combinados [6]. A variação da indutância é utilizada para estimativas em baixas velocidades, e Back-EMF para altas velocidades. Como a precisão deste sistema é muito afetada pelos ruídos das medições de tensão e corrente, muitas vezes são empregados filtros Kalman Extendidos (EKF, *Extended Kalman Filter*) [7]. Um controle SMC (Sliding Mode Control) pode ser utilizado com realimentação da corrente elétrica e posição do rotor [8]. Controles *sensorless* devem incluir o modelo do motor de passo e o algoritmo de controle, e isto torna o custo computacional alto para um microcontrolador PIC típico [9].

Os sistemas de controle de motor de passo, que usam sensores de posição, normalmente se baseiam em retardar a excitação das fases do motor até que ele tenha respondido satisfatoriamente à última excitação [10]. Assim, o motor não perde passos. A comutação entre as fases pode ser antecipada para gerar mais torque no eixo. Este ângulo entre o rotor e o que está sendo demandado eletricamente é chamado de *switching angle*, ou ângulo de controle. Ele pode ser otimizado em função do torque e velocidade no eixo do motor. Um controle ótimo pode ser obtido para melhor desempenho do motor de passo [11]. O ângulo de controle ótimo é obtido maximizando a equação do torque do motor de passo.

Entretanto, estas técnicas exigem que o sensor de posição seja diretamente acoplado ao eixo do motor, o que limita seu campo de aplicações, necessitando em muitos casos a alteração do projeto do manipulador ou a modificação do motor de passo. Outra desvantagem desta técnica ocorre quando o motor está acoplado a um redutor mecânico, sistema de polias ou um acoplamento elástico. Estes dispositivos dificultam a estimativa da posição do sistema em função da posição do eixo do motor, pois introduzem uma flexibilidade que induz um sistema não-colocado.

As limitações das técnicas para controle em malha fechada apresentadas motivaram este trabalho. O sistema desenvolvido neste trabalho, um manipulador robótico controlado por motores de passo, utiliza um *encoder* não diretamente acoplado ao motor, e sim ao elemento cuja posição será controlada. O algoritmo de controle não necessita de um alto custo computacional, sendo possível ser implementado em um microcontrolador PIC. Este trabalho propõe uma técnica de controle que recebe a realimentação de um *encoder* e gera uma sequência de pulsos para o *driver* do motor de passo. Esse trem de pulsos é feito de modo a não exigir acelerações excessivas, e assim prevenir a perda de passo. O controle proposto utiliza um controlador PID modificado. As características não lineares do motor de passo tornam o controle PID clássico ineficiente [12]. A partir de um modelo de motor de passo, uma simulação do controle proposto é feita usando o ambiente computacional Simulink/MATLAB. A estrutura deste modelo é replicada na simulação deste controle aplicado a um manipulador robótico que utiliza seis motores de passo. Este manipulador é projetado especialmente para esse trabalho, usando ferramentas de CAD (*Computer-Aided Design*). O modelo dinâmico do manipulador é incluído na simulação para incluir os torques gerados pelo seu movimento. A inércia dos elos, forças inerciais centrífugas e de Coriolis, e forças gravitacionais são incluídas no modelo.

Após a simulação demonstrar que o manipulador possui um desempenho adequado, quantificado no Capítulo 7, o manipulador é construído para validar a técnica de controle proposta. O circuito do controlador utiliza um microcontrolador PIC18F2431. Este controlador recebe a realimentação de *encoders* instalados nas juntas do manipulador. A trajetória do manipulador é gerada em um controlador CNC (Controle Numérico Computadorizado). Este controlador CNC possui uma interface na qual um código pode ser inserido. A linguagem utilizada é baseada em código G, que é largamente difundida na indústria, por ser utilizada em máquinas-ferramenta. Cada fabricante de manipulador robótico comercial possui sua própria linguagem [13], o que dificulta sua implementação na pequena e média indústria. O sistema de programação de trajetórias utilizado neste projeto apresenta esta vantagem sobre os comerciais. Outra vantagem do manipulador proposto é seu baixo custo, devido ao uso de motores de passo. Testes de precisão absoluta e repetibilidade são feitos neste manipulador.

Esta dissertação esta organizada em nove capítulos. O Capítulo 2 apresenta uma revisão do funcionamento dos motores de passo e técnicas de controle. No Capítulo 3 é feita a modelagem de motores de passo. O Capítulo 4 descreve a técnica de controle proposta. O Capítulo 5 apresenta em detalhes o sistema experimental desenvolvido. No Capítulo 6 o modelo do sistema experimental com o controle proposto é apresentado. O Capítulo 7 trata das simulações do sistema completo. O Capítulo 8 relata os resultados obtidos. E, finalmente, o Capítulo 9 apresenta a discussão dos resultados, as conclusões do trabalho e as propostas de trabalhos futuros.

2 Fundamentos teóricos

2.1. Motores de passo

Motores de passo são atuadores eletromecânicos incrementais não-lineares. Permitir um controle preciso de posição e velocidade, aliado a um baixo custo, faz do motor de passo a escolha ideal para um grande número de aplicações [14].

A propriedade que diferencia o motor de passo dos demais motores elétricos é a capacidade de fazer movimentos incrementais precisos em malha aberta. Essas rotações discretas são chamadas de passo [10].

As subseções abaixo apresentam os vários tipos de motores de passo e suas formas de excitação.

2.1.1. Motores de passo de relutância variável

Os motores de passo mais simples são os de relutância variável (VR, *Variable-Reluctance*). A Figura 1 mostra a seção esquemática de um motor deste tipo com seis dentes no estator.

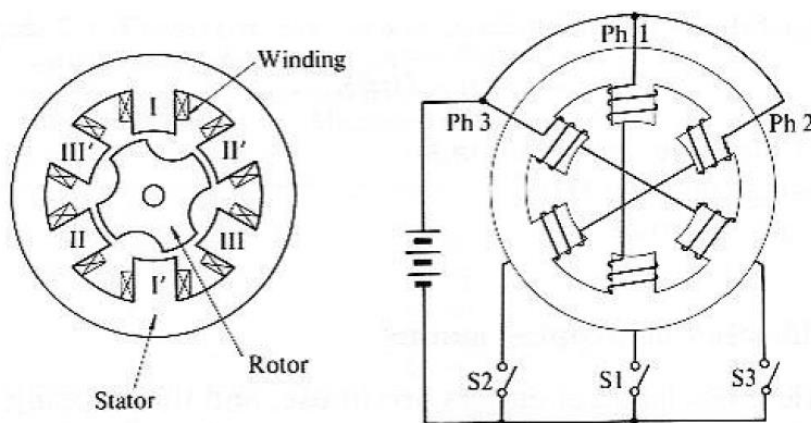


Figura 1 – Esquema motor VR

Na Figura 1, cada par de dentes opostos do estator estão ligados na mesma fase. O rotor possui quatro dentes. O estator e o núcleo do rotor normalmente são feitos em aço.

Se a corrente é aplicada na fase 1, *Ph1* na Figura 1, um campo magnético é criado e um dente do rotor tenta se alinhar a ele. A Figura 2 mostra como isso ocorre.

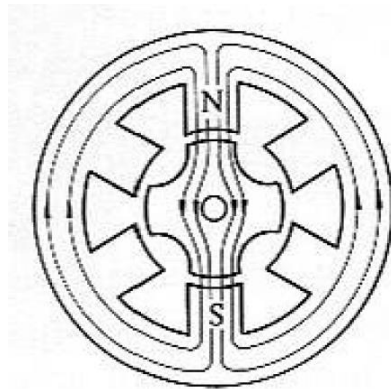


Figura 2 – Fluxo motor VR

Quando o rotor se alinha ao campo magnético, a relutância magnética é minimizada. Se o rotor se desalinhar do campo magnético, um torque é gerado no sentido de retornar à posição de equilíbrio.

2.1.2. Motores de passo de imã permanente

Estes motores de passo se caracterizam por seu rotor ser formado por um ímã permanente. O estator é formado por dentes envoltos por bobinas. Pode-se observar como é a estrutura básica de um motor de passo de quatro fases na Figura 3.

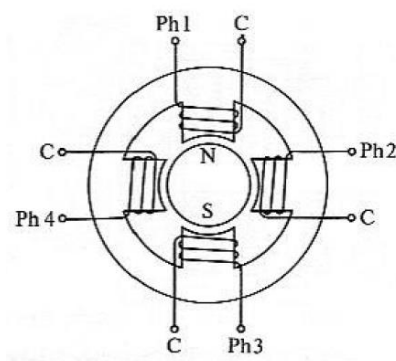


Figura 3 – Motor 4 fases

Ao energizar uma das fases, é criado um campo magnético. Assim, um torque é gerado pela atração do pólo gerado com o pólo oposto do ímã permanente. Um *driver* básico para controlar esse motor de passo é mostrado na Figura 4.

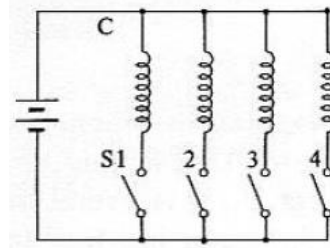


Figura 4 – Driver 4 fases

Se a chave “S1” estiver ligada, e depois ligarmos a chave “S2”, o motor irá efetuar uma rotação de 45°. A Figura 5 ilustra como isso é feito.

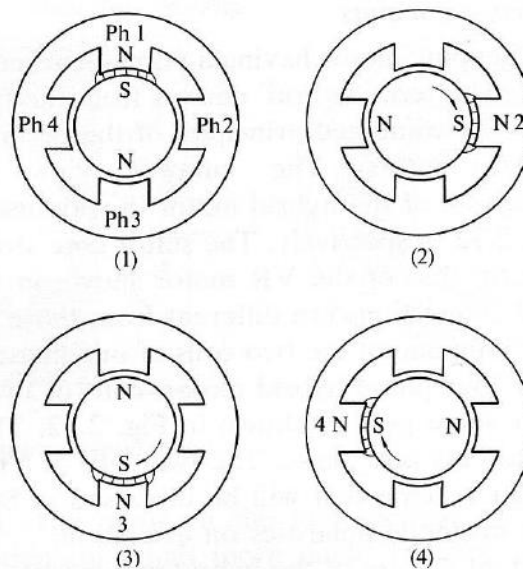


Figura 5 – Sequência motor de 4 fases

2.1.3. Motores de passo híbridos

Outro tipo de motor de passo com um ímã permanente em seu rotor é o híbrido. Esse tipo de motor é chamado de híbrido porque opera usando os princípios do motor de ímã permanente e de relutância variável [15]. O desenho esquemático de um motor de passo híbrido é mostrado na Figura 6.

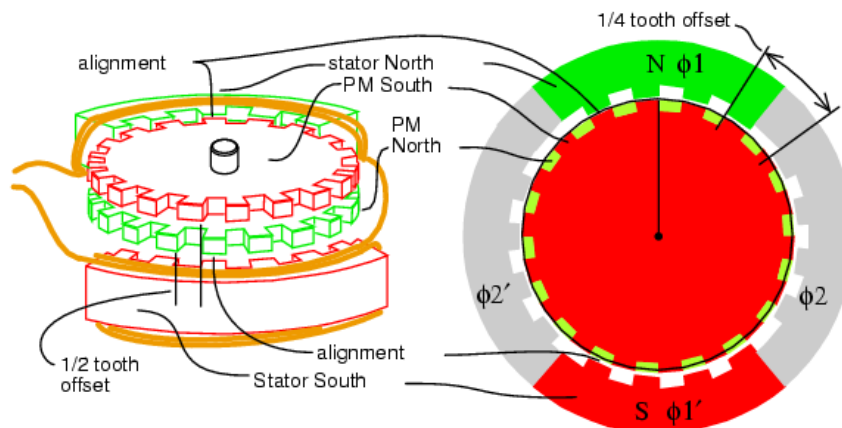


Figura 6 – Esquema motor híbrido

O estator deste motor é muito semelhante ao motor VR, mas os enrolamentos e as bobinas são conectados de forma diferente. No motor VR somente uma das duas bobinas de cada fase está em cada pólo. No motor híbrido, bobinas de duas fases diferentes estão no mesmo pólo. Ou seja, cada pólo não pertence a apenas uma fase. Desta forma, elas produzem polaridades magnéticas diferentes quando excitadas.

O núcleo do rotor do motor híbrido é permanentemente magnetizado, de modo a gerar um campo unipolar. Ao redor do rotor existe um “sanduíche” de engrenagens que são polarizadas pelo ímã do núcleo. Na Figura 6, a parte superior do rotor é o pólo sul e a parte inferior o norte. Os dentes dos pólos estão defasados de um dente.

2.1.4. Modos de excitação

A forma como as fases do motor de passo são excitadas determina o tamanho do incremento de rotação e o torque disponível. Existem três modos de excitação: passo completo, meio passo e micropasso. O modo passo completo pode ser obtido energizando as fases de forma sequencial, um ou duas de cada vez. Energizar duas fases de cada vez gera mais torque no rotor. Outra forma é energizar duas fases e desenergizar uma na sequência, obtendo assim um meio passo. No modo micropasso, é feita uma variação quase contínua da energização das fases – assim, um passo pode ser subdividido. O passo é dividido, tipicamente, por 4, 8 ou 16. Entretanto, existem sistemas capazes de subdividir um passo milhares de vezes. A desvantagem deste método é a maior complexidade do *driver* e menor precisão.

Motores híbridos permitem serem alimentados por um *driver* bipolar, assim os enrolamentos podem ser ligados de modo que o motor tenha duas fases, e cada fase duas polaridades. Desta forma, quando uma fase é acionada, metade dos enrolamentos do motor fica ligada. Em quanto que em motores unipolares, um quarto dos enrolamentos ficam ligados de cada vez. Isso permite ao motor híbrido ter até 50% mais eficiência e até 70% mais torque. A Figura 7 mostra a sequência de energização de um motor híbrido acionado por um *driver* bipolar.

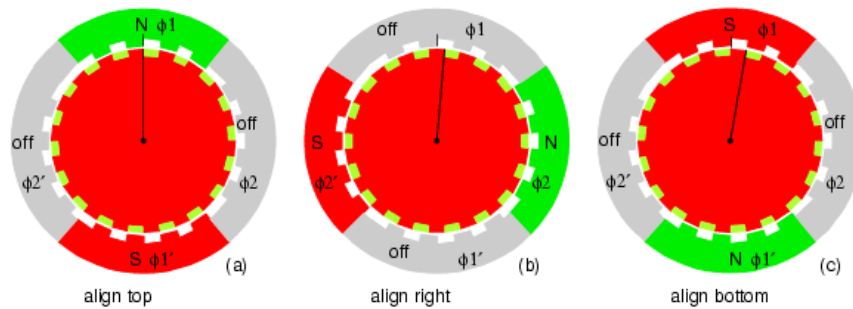


Figura 7 – Sequência motor híbrido

Na Figura 7-(a) o *driver* fornece um tensão positiva nos terminais da fase 1. Na Figura 7-(b) o *driver* fornece um tensão positiva nos terminais da fase 2, assim o motor realiza a rotação de um passo. Na Figura 7-(c) o *driver* fornece um tensão negativa nos terminais da fase 1, assim o motor realiza a rotação de mais um passo. A Figura 8 mostra como os terminais estão conectados ao estator.

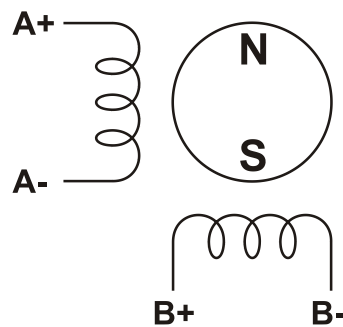


Figura 8 – Terminais de um motor com ligação bipolar

A Tabela 1 apresenta a sequência de polaridades que devem ser aplicadas aos terminais do motor para realizar passos no sentido horário. Para realizar passos no sentido anti-horário, basta seguir a tabela de baixo para cima.

Tabela 1 – Sequência para passo completo

Passo	A +	A -	B +	B -
0	+	-	+	-
1	-	+	+	-
2	-	+	-	+
3	+	-	-	+

A Tabela 2 apresenta a sequência de polaridades para o motor operar em modo de meio passo no sentido horário. O incremento na rotação é metade do modo passo completo.

Tabela 2 – Sequência para meio passo

Passo	A +	A -	B +	B -
0	+	-	+	-
1	-	-	+	-
2	-	+	+	-
3	-	+	-	-
4	-	+	-	+
5	-	-	-	+
6	+	-	-	+
7	+	-	-	-

A seção seguinte apresenta como o motor deve ser excitado em malha aberta.

2.2. Motores de passo em malha aberta

O motor de passo realiza rotações precisas a cada energização de seus enrolamentos. A sequência de excitação é feita pelo driver em função de um conjunto de sinais de entrada. Na maioria das aplicações é utilizado um sinal chamado passo e outro chamado direção. O sinal de direção determina o sentido da rotação do motor, horário ou anti-horário. A cada pulso no sinal de passo, o motor deve realizar a rotação de um incremento.

Como o valor do incremento é conhecido, a partir da contagem dos pulsos do sinal passo, a posição do rotor pode ser determinada. Isto é possível desde que

o motor tenha torque disponível para realizar esta rotação. A partir da frequência dos pulsos de passo, a velocidade do motor pode ser estimada.

A Figura 8 mostra entradas de passo e direção típicas. O *driver*, que recebe sinais como os da figura, demanda ao motor realizar dois incrementos de rotação em um sentido e três no outro.

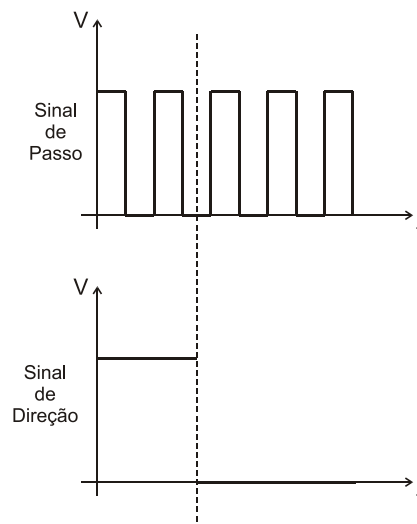


Figura 8 – Sinais de comando

Esses incrementos de rotação são muito precisos. Essa característica de precisão em malha aberta é uma grande vantagem para os motores de passo. Porém, se o torque aplicado ao eixo exceder o limite do motor, haverá um erro permanente no posicionamento do rotor. Então motores de passo em malha aberta devem ser super-dimensionados. Como o torque disponível é significativamente reduzido em altas rotações, a velocidade máxima do motor deve ser limitada a um nível de segurança. Ou seja, para assegurar que o motor execute as rotações esperadas em função da excitação, o seu desempenho deve ser sacrificado. As limitações do controle em malha aberta estimularam o desenvolvimento do controle em malha fechada, apresentado na seção seguinte.

2.3. Motores de passo em malha fechada

A forma mais comum de controle de motor de passo em malha fechada utiliza a realimentação da posição do eixo do rotor [11]. Cada pulso no sinal de

passo só é realizado se o motor respondeu adequadamente ao ultimo comando [10]. A Figura 9 mostra a malha de controle deste sistema.

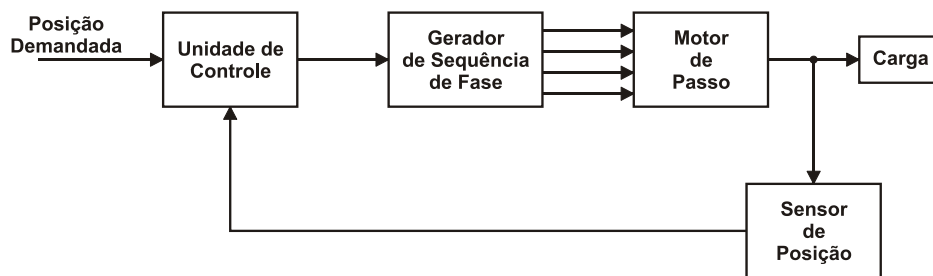


Figura 9 – Malha fechada típica

Esta técnica se baseia em descobrir quando energizar cada fase, em função da posição do rotor, para maximizar o torque no eixo. Em baixas velocidades, o ponto ótimo para comutar entre as fases pode ser deduzido de uma análise estática do torque gerado por cada fase. A Figura 10 apresenta um gráfico do torque gerado por cada fase em função da posição do rotor, para um motor de três fases.

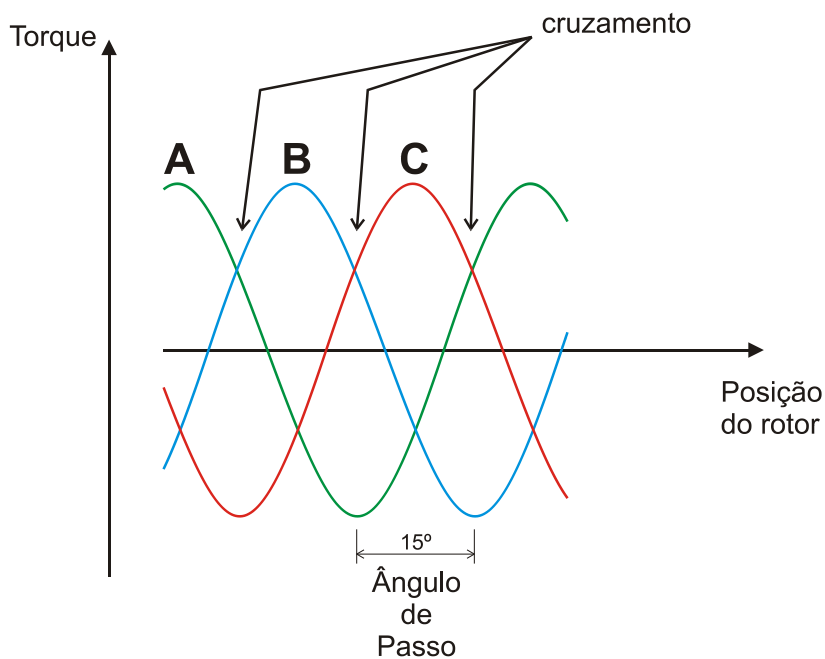


Figura 10 – Torque vs. posição por fase

Em baixa velocidade, o torque pode ser maximizado comutando entre as fases no ponto de cruzamento das curvas de torque. Mas, em alta velocidade, os efeitos indutivos das bobinas do motor geram distorções na forma de onda da corrente. Nessas condições, a comutação deve ocorrer antes para dar tempo da corrente que circula a fase se estabilizar. O ângulo ótimo de comutação pode ser

extraído de parâmetros construtivos do motor. A Figura 11 apresenta a relação típica entre torque e velocidade, para vários ângulos de comutação.

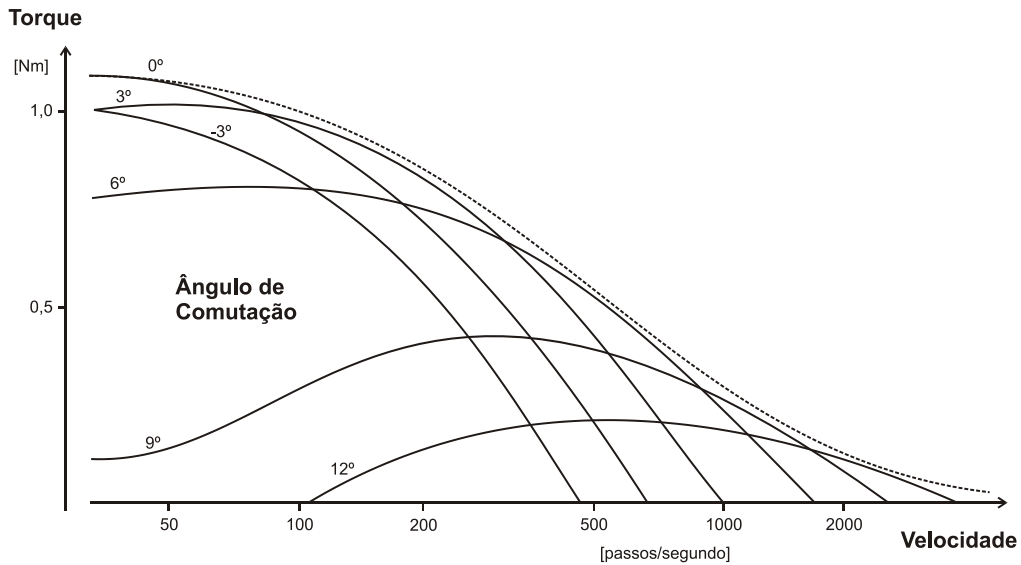


Figura 11 – Torque vs. velocidade para vários ângulos de comutação

Como os parâmetros do motor são constantes, cada velocidade de operação possui um ângulo de comutação ótimo. Logo, para se obter o máximo torque, o controlador deve ser capaz de variar continuamente o ângulo de comutação, e isso depende de uma leitura contínua da posição do rotor. Na prática, são suficientes alguns valores discretos. Por exemplo, um motor com resolução de 200 passos por volta necessita de um sensor com 1600 pontos por volta, resultando em 8 possíveis ângulos de comutação.

Esta técnica exige que o sensor seja montado diretamente no eixo do motor. Normalmente isso é feito fixando um *encoder* em uma extensão traseira do eixo do rotor. Entretanto, os fabricantes de motores de passo disponibilizam poucos modelos de motores com essa opção. A Figura 12 mostra um motor de passo com *encoder* acoplado.



Figura 12 – Foto de um motor com *encoder*

Além dos modelos onde *encoders* podem ser facilmente adaptados serem poucos, o volume do conjunto também é maior. Logo, em aplicações onde o compartimento do motor é pequeno, pode ser difícil acondicionar motor e *encoder* juntos. Outra limitação para esta técnica pode ser observada em aplicações onde a posição de interesse para o controle não é a do rotor. Por exemplo, em robótica é comum o eixo do motor ser acoplado a uma caixa redutora de velocidade. No eixo de saída desta caixa redutora, um elemento como a junta de um manipulador serial pode ser acoplada. Se a caixa do redutor possuir folga (*backlash*), a posição da junta do manipulador não pode ser conhecida apenas a partir da posição do eixo do motor. Em uma aplicação como esta, um *encoder* montado na saída da caixa de redução permite compensar o erro devido à folga do redutor. A técnica de controle, usando o ângulo de comutação, não poderia ser usada. Este fenômeno da folga é apresentado na seção seguinte.

2.4. Redutores com folga (*backlash*)

Backlash é a não linearidade mais importante em acionamentos industriais e grande causa da deterioração do controle de posição [16]. Em muitas aplicações, uma caixa redutora de velocidade é acoplada ao motor de passo para aumentar o torque de saída. A caixa de redução é composta por um conjunto de engrenagens. Devido a limitações no processo de fabricação, cada par de engrenagens possui uma pequena folga, ou *backlash* [17]. A Figura 13 mostra, esquematicamente, a folga entre um par de engrenagens.

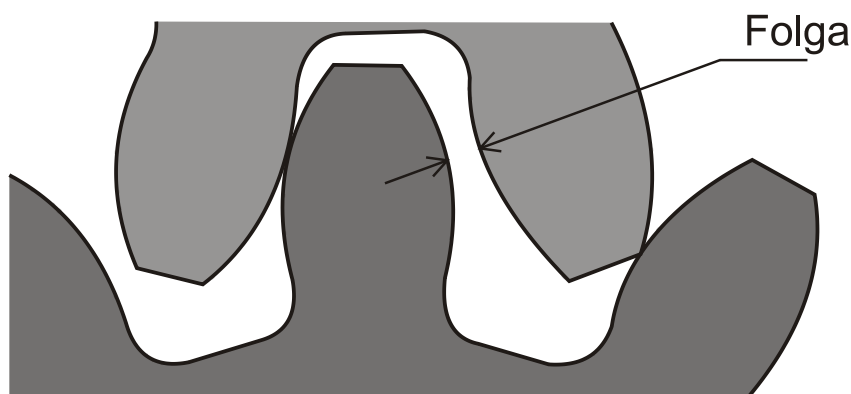


Figura 13 – Folga entre dentes

O erro devido ao *backlash* é difícil de estimar e compensar porque não pode ser descrito como uma relação linear [18]. A Figura 14 mostra a variação do ângulo de saída devido ao *backlash*.

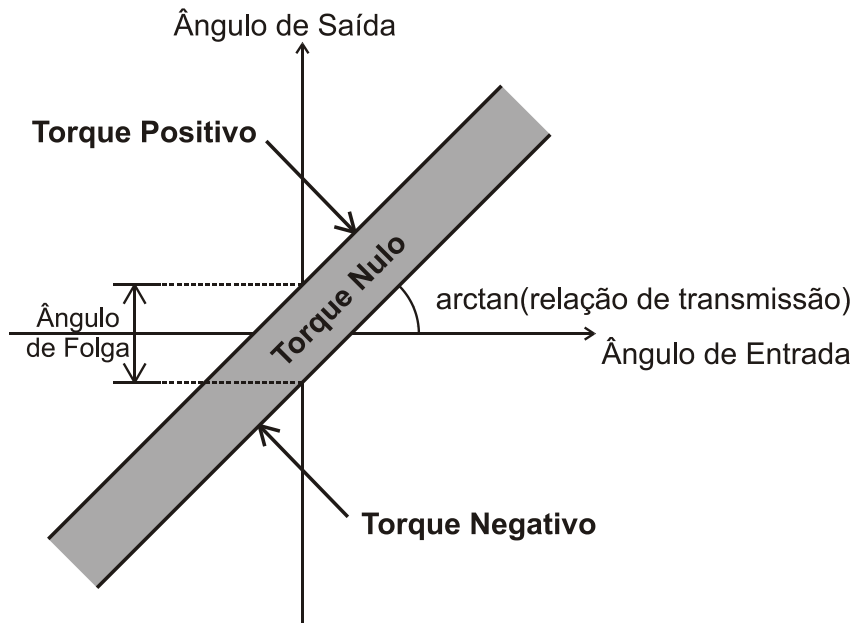


Figura 14 – Ângulo do eixo de entrada vs. saída com folga

A posição do eixo de saída pode ser conhecida se houver torque no eixo de saída, e se seu sentido for conhecido. Se o torque for nulo, para cada posição do eixo de entrada existe uma faixa de valores possíveis para a posição do eixo de saída. A inclinação do gráfico depende da relação de transmissão, no caso de um redutor de engrenagens, do número de dentes de cada engrenagem.

Backlash normalmente é modelado como uma zona morta. Nesta modelagem, o torque é dado em função da diferença entre os ângulos de saída e entrada [16]. Entretanto, essa modelagem exige que o atrito interno seja desprezível. Outro modelo de *backlash* propõe a modelagem através de uma histerese, onde a posição dentro da região de folga é dada em função do histórico no tempo da posição do eixo.

No próximo capítulo, a modelagem de motores de passo híbridos é apresentada.

3 Modelagem de motores de passo

Neste capítulo é estudado um modelo de motor de passo híbrido. O modelo desenvolvido é implementado no ambiente computacional Simulink/MATLAB. Este modelo pode ser utilizado em motores de imã permanente, apesar de ser estruturalmente diferente do híbrido [15]. Uma análise de como o motor de passo é capaz de gerar torque é apresentada. A teoria do comportamento dinâmico apresentado é baseada no modelo proposto por Lawrenson e Hughes [19]. O modelo é baseado em um modelo elétrico e um mecânico. Estes dois modelos estão conectados pelo torque que aparece em ambos. No elétrico, aparece o torque gerado pela força eletromagnética. No mecânico, o torque aparece na equação de movimento de um rotor com inércia apoiado em mancais com atrito.

Uma modelagem mais precisa dos motores de passo exige um modelo em elementos finitos tridimensional do seu interior [20]. Estes modelos são utilizados por fabricantes para otimizar o desempenho dos motores de passo. A modelagem apresentada a seguir é mais simples, baseada em um circuito equivalente linear. Este modelo é suficiente para avaliação de algoritmos de controle [21]. A seguir é apresentado o circuito equivalente deste modelo.

3.1. Circuito equivalente

O circuito equivalente apresentado se baseia em um circuito RL (Resistor-Indutor) com uma fonte de tensão adicional. Este circuito não considera a indutância mútua entre as fases do motor e a saturação do circuito. A Figura 15 mostra o circuito equivalente da fase A.

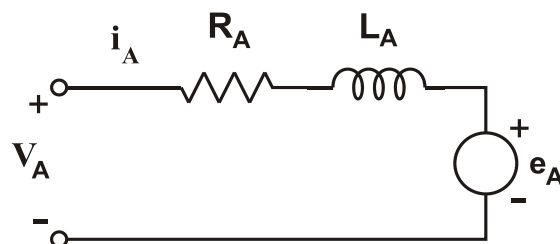


Figura 15 – Circuito equivalente

Nesta figura, R_A é a resistência equivalente da fase A, L_A a indutância, v_A a tensão nos terminais e e_A a tensão induzida. Esta tensão induzida é gerada devido à variação do fluxo magnético com o movimento do imã permanente. O fluxo induzido varia de acordo com o cosseno do ângulo do rotor. Sendo um motor de duas fases, o fluxo induzido, em cada uma das fases, pode ser representado por

$$\Psi_A = \Psi_M \cos(p\theta) \quad (3.1)$$

$$\Psi_B = \Psi_M \cos(p\theta - \pi/2) \quad (3.2)$$

onde p é o número de dentes do rotor, θ o ângulo de rotação e Ψ_M é o fluxo induzido máximo. Neste caso é considerado que quando $\theta=0$ o pólo norte do rotor está alinhado com a fase A. A tensão induzida e_A é igual à taxa de variação do fluxo induzido. No caso de um rotor girando com velocidade $\frac{d\theta}{dt}$, as tensões

induzida nas fases A e B são dadas por

$$e_A = \frac{d\Psi_A}{dt} = -p\Psi_M \text{sen}(p\theta) \frac{d\theta}{dt} \quad (3.3)$$

$$e_B = \frac{d\Psi_B}{dt} = -p\Psi_M \text{sen}(p\theta - \pi/2) \frac{d\theta}{dt} \quad (3.4)$$

A equação da tensão no terminal do circuito equivalente da fase A é

$$v_A = R_A i_A + L_A \frac{di_A}{dt} + e_A \quad (3.5)$$

Substituindo a eq. (3.3) na eq. (3.5), a equação da tensão nos terminais da fase A é obtida:

$$v_A = R_A i_A + L_A \frac{di_A}{dt} - p\Psi_M \text{sen}(p\theta) \frac{d\theta}{dt} \quad (3.6)$$

Analogamente, a equação da tensão da fase B é

$$v_B = R_B i_B + L_B \frac{di_B}{dt} - p\Psi_M \text{sen}(p\theta - \pi/2) \frac{d\theta}{dt} \quad (3.7)$$

A seguir é apresentado um modelo para o torque gerado.

3.2. Torque gerado

O torque gerado em um motor de passo é resultado de dois efeitos: variação da relutância ao longo da rotação do rotor e o torque eletromagnético gerado pelo imã permanente. Entretanto, em motores híbridos, o torque eletromagnético é

dominante [21]. No modelo proposto, o torque gerado pela variação da relutância é desprezado. A Figura 16 apresenta o esquema de um motor híbrido.

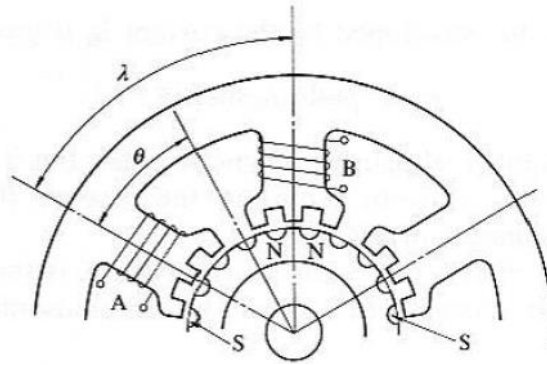


Figura 16 – Esquema ímã híbrido

Neste modelo, o ângulo entre as fases está sendo considerado $\lambda = \pi/2$. Como os circuitos magnéticos são considerados lineares, e a indução entre as duas fases é desprezível, é assumido que o torque total é a soma dos torques gerados por cada uma das fases [15]. A partir disto, o torque gerado pela corrente i_A no enrolamento A é dado por

$$\tau_A = -p\Psi_M i_A \sin(p\theta) \quad (3.8)$$

Devido à fase B estar defasada da fase A, o torque gerado pode ser descrito por

$$\tau_B = -p\Psi_M i_B \sin(p\theta - \pi/2) \quad (3.9)$$

Além destes torques gerados, existe o torque devido à interação dos dentes do estator com os pólos magnéticos do rotor. Este torque é chamado de torque de retenção T_{dm} , ou *detent torque*. Este efeito faz os motores de passo apresentarem torque magnético de retenção mesmo quando desligados. Tipicamente, o torque de retenção está entre 1% e 10% do torque máximo.

Somando os efeitos dos torques gerados na fase A, na fase B e o torque de retenção, temos para o torque eletromagnético gerado:

$$\tau_e = -p\Psi_M [i_A \sin(p\theta) + i_B \sin(p\theta - \pi/2)] - T_{dm} \sin(2p\theta) \quad (3.10)$$

A seguir é apresentado o efeito do torque no movimento do motor.

3.3. Equação de movimento do motor

O torque gerado pelo motor pode ser acoplado a uma carga na extremidade. Este torque gerado também é utilizado para acelerar o rotor e a carga. Além disso,

o atrito dos mancais do motor e da carga dissipa parte da potência gerada pelo motor. Logo, a equação de movimento deste sistema é dada por

$$\tau_e = J \frac{d^2\theta}{dt^2} + D \frac{d\theta}{dt} + T_L \quad (3.11)$$

onde:

D - Coeficiente de atrito viscoso total (motor+carga).

J - Inércia total (motor+carga).

T_L - Torque externo aplicado.

As equações dos terminais das fases, (3.6) e (3.7), junto com as equações do torque, (3.10) e (3.11), são utilizadas para modelar o motor de passo. Devido à sua complexidade, a simulação do modelo é feita usando o *software* Simulink/MATLAB para a solução numérica. O modelo utilizado considera o circuito equivalente RL (Resistor-Indutor) linear, ou seja, sem saturação. Algumas perdas de fluxo magnético também são desconsideradas.

3.4.

Relação entre dentes, fases e número de passos

Os dentes no interior dos motores de passo são necessários para gerar torque e posicionar o rotor. A Figura 17 mostra como os dentes de um motor de passo são arranjados no rotor e no estator.

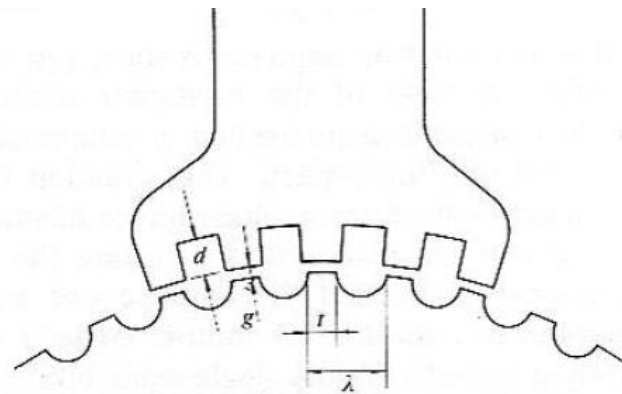


Figura 17 – Dentes rotor e estator

O número de passos por revolução, quando os dentes no estator e no rotor têm o mesmo passo de fabricação, é dado por

$$S = mN_r \quad (3.12)$$

onde:

m – Número de fases

N_r – Número de dentes do rotor; em um motor híbrido, $N_r = p$.

Assim, o ângulo de passo, i.e. incremento na rotação feito por cada passo completo, é dado por

$$\theta_s = \frac{360^\circ}{S} \quad (3.13)$$

3.5. Cálculo do fluxo induzido máximo

O modelo do motor de passo exige o fluxo eletromagnético máximo induzido. Entretanto, este parâmetro dificilmente é fornecido pelos fabricantes de motores de passo. Este parâmetro pode ser calculado girando o motor com velocidade constante e medindo a tensão nos terminais de uma das fases. Como nesta situação o circuito está aberto, não há corrente na fase, logo a eq. (3.6) pode ser reescrita por

$$v_A = -p\Psi_M \text{sen}(p\theta) \frac{d\theta}{dt} \quad (3.14)$$

A partir desta equação, a tensão máxima medida nos terminais da fase A vale

$$E_M = p\Psi_M \dot{\theta} \quad (3.15)$$

Provavelmente seria utilizado um tacômetro com escala em rotações por minuto (RPM) para medir a velocidade angular. Logo, é interessante colocar a expressão de Ψ_M em função de N , com unidade de RPM. A expressão utilizada para o cálculo de Ψ_M em função dos valores medidos resulta então em

$$\Psi_M = \frac{30E_M}{p\pi N} \quad (3.16)$$

No próximo capítulo, o controle dos motores de passo é discutido.

4 Controle de motores de passo

O controle em malha aberta é o mais comum em motores de passo. Entretanto, o motor deve operar razoavelmente abaixo de sua capacidade para evitar a perda de passos. As limitações deste tipo de controle estimularam o desenvolvimento de técnicas de controle em malha fechada (MF) [22]. A forma tradicional de controle em MF utiliza um *encoder* diretamente acoplado ao eixo do motor, como dito no Capítulo 2, mas nem sempre isto é tecnicamente viável. Algumas novas técnicas, sem a necessidade de um sensor de posição, surgiram nos últimos anos [23]. Essas técnicas, chamadas *sensorless*, não apresentam ainda suficiente confiabilidade para algumas aplicações. Neste capítulo é proposta uma técnica baseada no controle da posição de um eixo ligado ao motor por um sistema de transmissão que pode apresentar perturbações ou flexibilidades. Ou seja, a posição do eixo do motor não pode ser determinada com precisão a partir do sensor de posição. A seção abaixo apresenta o controle proposto para aplicação em motores de passo.

4.1. Malha de controle

A Figura 18 apresenta a malha de controle proposta. Nesta malha, como em malha aberta, um *driver* recebe pulsos de passo e direção e gera corrente nas fases do motor de passo. Este motor de passo é ligado a um redutor de velocidade mecânico. Este redutor é ligado a um *encoder* através de um par de polias.

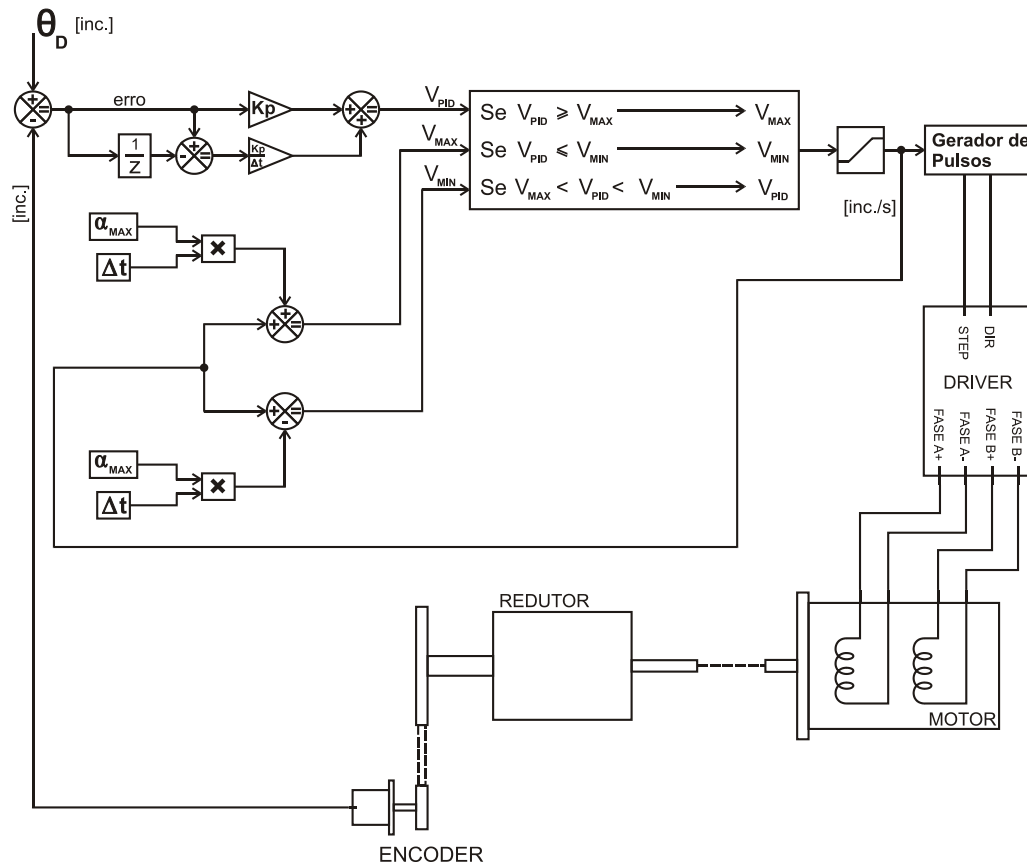


Figura 18 – Malha de controle proposta

A velocidade demandada ao motor depende da frequência dos pulsos de passo (*STEP*) na entrada do *driver*. Em malha aberta, essa frequência pode ser variada para gerar uma determinada aceleração [3, 24].

Neste trabalho, a frequência dos pulsos de passo enviados para *driver* é determinada por um gerador de pulsos, que envia também um sinal de direção. Este gerador de pulsos recebe na entrada a velocidade demandada para o motor e calcula a frequência de pulsos de passos. O gerador de pulsos calcula essa frequência através da relação

$$f = \frac{\omega}{\theta_s} \quad (4.1)$$

onde:

f - frequência dos pulsos de passo;

ω - velocidade angular do motor.

O valor da velocidade na entrada do gerador de pulsos corresponde à quantidade de passos por segundo que o controlador deve gerar. Logo, a saída do

controlador V_C também possui esta mesma escala. Assim, o valor do parâmetro de aceleração α utiliza a unidade de *passos/segundo*².

O sinal de direção (DIR) é um binário com nível alto se a velocidade demandada no gerador é positiva, e nível baixo se é negativa.

A variável de entrada do gerador de pulsos – velocidade - é calculada a partir de um controlador. O controlador gera um perfil de velocidades suave, para prevenir a perda de passo no motor. Isto é feito calculando as velocidades máxima e mínima na saída que não gerem uma aceleração maior que a máxima. Esta aceleração máxima é constante, e é um dos parâmetros que devem ser ajustados quando o controlador é implementado em um sistema.

O controlador calcula também uma velocidade usando uma lei de controle proporcional-derivativa (PD). O erro é calculado pela diferença entre a posição desejada e a medida pelo *encoder*. Como o sistema é discreto, a derivada no tempo deste erro é aproximada por $\frac{\text{erro}(k) - \text{erro}(k-1)}{\Delta t}$. O erro é multiplicado por um ganho proporcional (K_p), a derivada do erro é multiplicada por um ganho derivativo (K_D), e assim é calculada a saída do algoritmo PD [25].

O controlador possui três velocidades calculadas: velocidade pela lei de controle PD (V_{PD}), velocidade mínima que não ultrapassa máxima desaceleração (V_{MIN}) e velocidade máxima que não ultrapassa máxima aceleração (V_{MAX}). O controlador decide, entre elas, qual a saída (V_C) através de uma base de regras. A seção seguinte apresenta esta base de regras.

4.2. Base de regras

O perfil de velocidade demandado ao motor deve ser suave, para prevenir a perda de passo. A base de regras é formulada para evitar variações exageradas da velocidade na saída.

A base de regras utilizada é apresentada abaixo.

Se $V_{PD} \geq V_{MAX}$, então $V_C = V_{MAX}$.

Se $V_{PD} \leq V_{MIN}$, então $V_C = V_{MIN}$.

Se $V_{MIN} < V_{PD} < V_{MAX}$, então $V_C = V_{PD}$.

onde a velocidade mínima é calculada por

$$V_{MIN}(k) = V_C(k-1) - \alpha \cdot \Delta t \quad (4.2)$$

onde Δt é o período de controle. A velocidade máxima é calculada por

$$V_{MAX}(k) = V_C(k-1) + \alpha \cdot \Delta t \quad (4.3)$$

e a velocidade associada ao PD é calculada por

$$V_{PD}(k) = erro(k) \cdot K_p + \frac{erro(k) - erro(k-1)}{\Delta t} \cdot K_D \quad (4.4)$$

O torque em um motor de passo é inversamente proporcional à velocidade de rotação do rotor. A partir de um determinado limite, o torque disponível é muito baixo e o motor tende a perder passos. Então, a velocidade V_C também é limitada para não ultrapassar um determinado limite. O limite é dado pela velocidade de saturação da saída do controlador (V_S). Na seção seguinte é proposta uma rotina para estimar as constantes do controlador.

4.3. Estimativa das constantes do controlador

O controlador precisa que quatro constantes sejam fornecidas: α , V_S , K_p e K_D . A constante V_S pode ser estimada do gráfico de desempenho do motor, α por um ensaio simples do sistema, enquanto que K_D pode ser inicialmente estimado como um valor muito baixo, e ajustado conforme o comportamento do sistema. Finalmente, K_p pode ser estimado com base nos resultados dos ensaios.

Primeiramente, é recomendado estimar V_S a partir do gráfico de desempenho do motor de passo. O torque máximo disponível é inversamente proporcional à velocidade angular. Assim, sabendo o torque máximo demandado ao motor, a velocidade máxima pode ser extraída do gráfico de desempenho. A Figura 19 apresenta o gráfico de desempenho de um motor típico. Sabendo o torque necessário, basta obter no gráfico a velocidade máxima V_S (em *passos/s*).

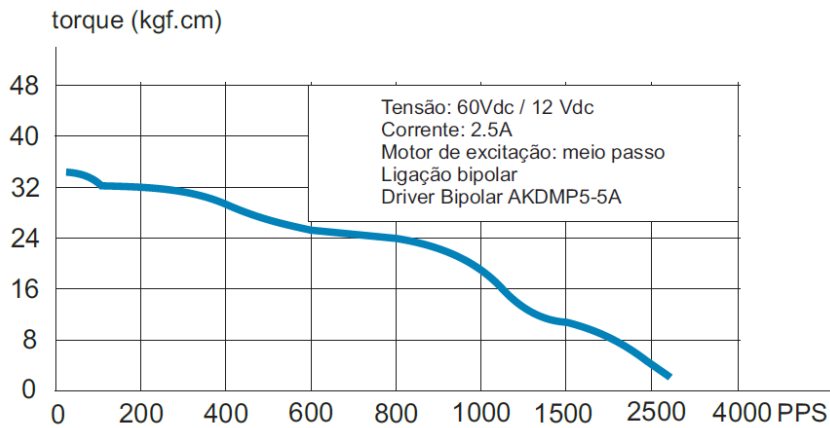


Figura 19 – Gráfico de desempenho de um motor de passo

A seguir, é proposto um ensaio para determinar o parâmetro α . Este parâmetro está relacionado à aceleração do sistema. Assumindo a constante K_p alta, e aplicando um degrau significativo na posição pelo controlador, a aceleração do motor é próxima a α . Assim, ao demandar um degrau na posição do sistema, o valor de saída do controlador depende inicialmente do valor de α . Este valor pode ser estimado como “baixo” inicialmente, e pode ser elevado gradativamente. Em algum momento a aceleração se torna excessiva e o motor começa a perder passos de forma contínua, logo o valor de α deve ser o valor anterior a este. Se o motor perder passos apenas a partir de uma velocidade alta, a constante V_s estimada pode ser reduzida.

Os valores de α e V_s encontrados podem ser usados para estimar a constante K_p . Se o motor for capaz de desacelerar de forma simétrica de como acelera, as constantes podem ser estimadas de modo a obter este resultado. A Figura 20 mostra o gráfico esquemático do erro da posição de um motor de passo com desaceleração constante.

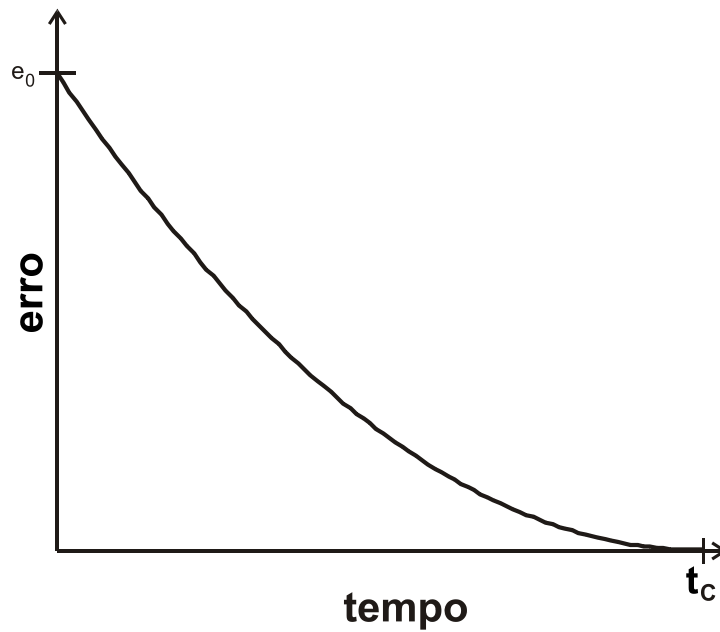


Figura 20 – Gráfico esquemático erro vs. tempo de um sistema

Neste gráfico, no instante inicial o motor está com velocidade máxima e o erro é e_0 . Nos instantes seguintes, ele desacelera constantemente até parar no instante t_c , com erro nulo. Para que o erro do sistema se comporte desta forma, o controlador deve demandar um perfil de velocidade linear ao motor. A Figura 21 mostra o perfil de velocidade para esta condição.

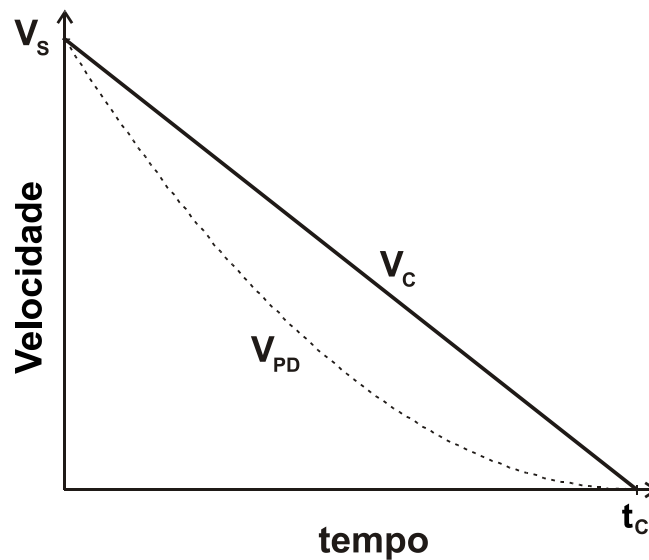


Figura 21 – Desaceleração constante do motor

Neste gráfico, o motor idealmente possui a velocidade exata que o controlador está demandando. Isto é razoável, desde que o motor não esteja perdendo passos. A linha V_{PD} mostra o valor da saída calculada pelo algoritmo

PD. O controlador utiliza V_{PD} em apenas dois instantes, $t = 0$ e $t = t_c$. Entre estes dois instantes tem-se $V_{PD} \leq V_{MIN}$, logo pela base de regras calcula-se $V_C(k) = V_{MIN}(k) = V_C(k-1) - \alpha \cdot \Delta t$. Desse modo, o sistema deve gerar uma desaceleração constante igual a α . No instante inicial, a velocidade é máxima (V_S), e no instante final é nula. Logo, o erro em função do tempo pode ser determinado por

$$e(t) = \frac{\alpha \cdot t^2}{2} - V_S \cdot t + e_0 \quad (4.5)$$

e a derivada do erro com relação ao tempo é

$$\dot{e}(t) = \alpha \cdot t - V_S \quad (4.6)$$

No instante final, $t = t_c$, a derivada do erro é nula, pois a velocidade é igual a zero. Substituindo isso na eq. (4.6), obtém-se para o instante t_c :

$$t_c = \frac{V_S}{\alpha} \quad (4.7)$$

Substituindo este valor de t_c na eq.(4.5), o erro deve ser nulo. A partir disto, obtém-se o erro no instante inicial

$$e_0 = \frac{V_S^2}{2 \cdot \alpha} \quad (4.8)$$

No caso de um sistema contínuo, a lei de controle PD é dada por

$$V_{PD}(e(t)) = e(t) \cdot K_P + \dot{e}(t) \cdot K_D \quad (4.9)$$

onde V_{PD} no instante inicial deve ser igual a V_S . E, no instante inicial, $e(t=0) = e_0$. Substituindo isso na eq. (4.9), obtém-se o erro no instante inicial em função dos ganhos e da velocidade de saturação:

$$e_0 = V_S \cdot \frac{(1 + K_D)}{K_P} \quad (4.10)$$

Igualando as eqs. (4.8) e (4.10), obtém-se

$$\frac{(1 + K_D)}{K_P} = \frac{2 \cdot \alpha}{V_S} \quad (4.11)$$

Supondo um ganho derivativo $K_D \approx 0$, obtém-se um valor aproximado do ganho K_P :

$$K_p \cong \frac{2 \cdot \alpha}{V_s} \quad (4.12)$$

Esta equação relaciona os parâmetros obtidos pelo ensaio proposto, V_s e α , com a constante K_p . O valor da constante K_D inicialmente deve ser estipulado como um valor “pequeno”. A partir deste ponto, é esperado que controlador possa controlar a posição do motor de forma eficiente, sendo necessários apenas pequenas alterações dos parâmetros para um ajuste fino.

No capítulo seguinte, o sistema experimental utilizado para validar as técnicas de controle propostas é apresentado.

5 Sistema experimental

O sistema experimental deste trabalho tem o objetivo de analisar o desempenho de motores de passo com o controle proposto. Isso é implementado em um sistema com necessidades críticas: um manipulador robótico serial. Erros de posicionamento pequenos nas juntas dos manipuladores podem deteriorar muito a precisão da extremidade. Além disso, manipuladores de porte industrial precisam de motores de passo acoplados a redutores de velocidade. Os motores comercialmente disponíveis não dispõem de torque suficiente para acionamento direto. Entretanto, estes redutores normalmente tornam a relação entre o ângulo da junta e o do motor não linear, em função do *backlash*.

Devido exigências críticas de desempenho, um manipulador robótico serial com seis graus de liberdade é escolhido para avaliar o controle proposto. A seguir é apresentado o projeto deste manipulador.

5.1. Projeto do manipulador

O manipulador proposto possui seis graus de liberdade, alcance de aproximadamente $560mm$, e capacidade de carga de $6kg$. O manipulador é proposto com essas características porque elas são comuns a manipuladores comerciais de pequeno porte que não usam motores de passo. A Figura 22 mostra uma imagem gerada pelo programa CAD, utilizado para desenvolver o manipulador.

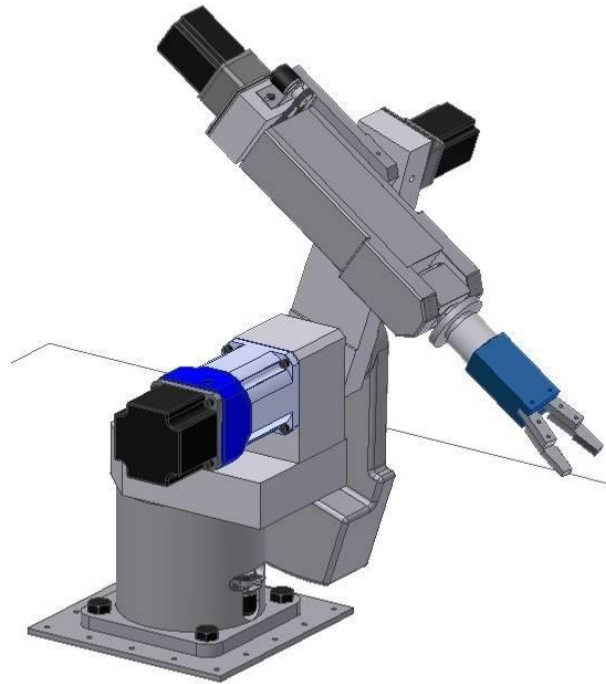


Figura 22 – Manipulador robótico

O projeto do manipulador visa gerar as maiores forças possíveis na sua extremidade, respeitando as limitações dos motores disponíveis. Então, para estudar quais termos da dinâmica do manipulador possuem relevância sobre o torque em cada junta, deve-se estudar a combinação dos efeitos:

$$Inercial + Centrífugo + Coriolis + Gravitacional = Torque \quad (5.1)$$

O primeiro termo desta equação depende do momento de inércia de cada elo e da aceleração angular. Então, para reduzir os efeitos deste termo, o manipulador deve ter elos com mínimos momentos de inércia. Neste manipulador, a aceleração máxima de cada elo é um parâmetro que deve ser informado ao controlador.

Os termos centrífugos e de Coriolis são proporcionais à velocidade, e dependem da configuração e da massa dos elos. Porém, como motores de passo não geram grandes velocidades, estes termos têm menor influência na avaliação do desempenho.

Os termos gravitacionais são resultado do torque gerado pelo peso dos elos, que depende da projeção horizontal da distância do CG de cada elo até cada uma das juntas. Então, para reduzir o efeito da gravidade sobre o manipulador, o seu projeto visa localizar o centro de gravidade de cada elo o mais próximo possível do eixo de cada junta. Deste modo, a força que pode ser gerada na extremidade é

maximizada, pois uma fração pequena do torque dos motores é utilizada para compensar a gravidade.

A Figura 23 contém a nomenclatura de alguns componentes do manipulador desenvolvido. Nesta figura aparecem os seis motores e os seis elos.

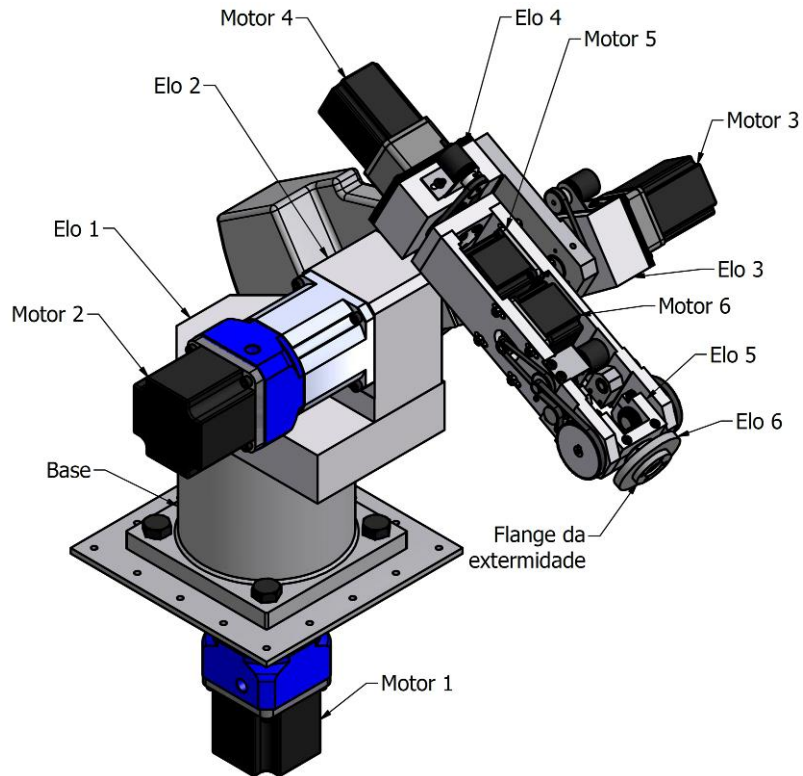


Figura 23 – Localização dos motores

5.1.1. Características principais

A Figura 24 contém as dimensões básicas do manipulador, em mm.

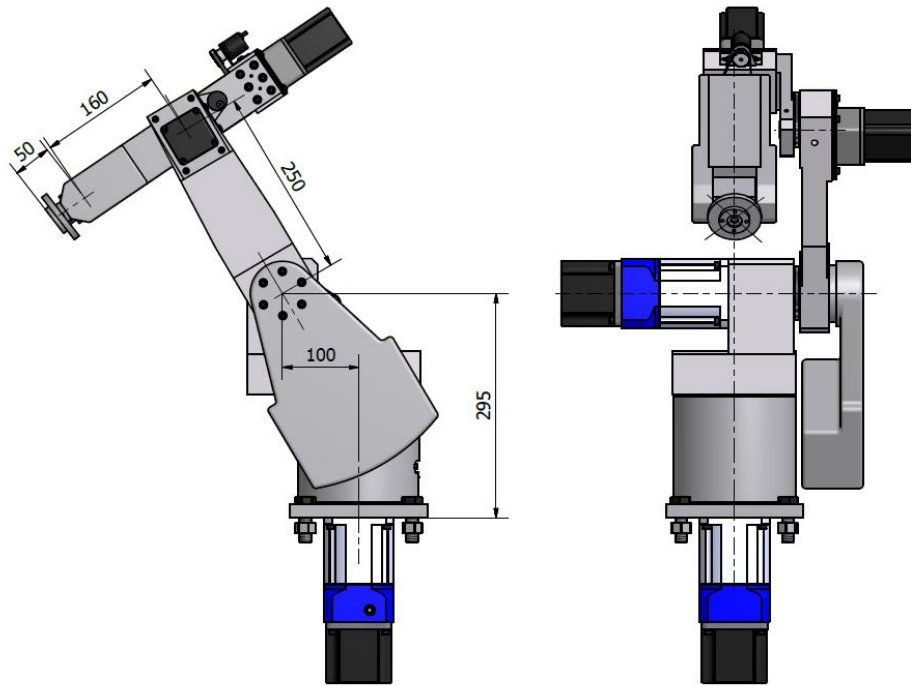


Figura 24 – Dimensões básicas

Através desta figura, é calculado que, se o manipulador estiver totalmente distendido, ele alcança 560mm na horizontal em relação ao eixo do primeiro motor. Porém, nesta configuração, o motor da junta 4 colide com a junta 2, logo este manipulador não pode ser completamente distendido. Assim o máximo alcance real do manipulador no flange da extremidade é de cerca de 540mm .

O atrito de cada junta é minimizado através de uma montagem com rolamentos de esfera. Na junta 1 são utilizados dois rolamentos de contato angular e um sistema de pré-carga. A Figura 25 mostra uma vista explodida das três primeiras juntas do manipulador.

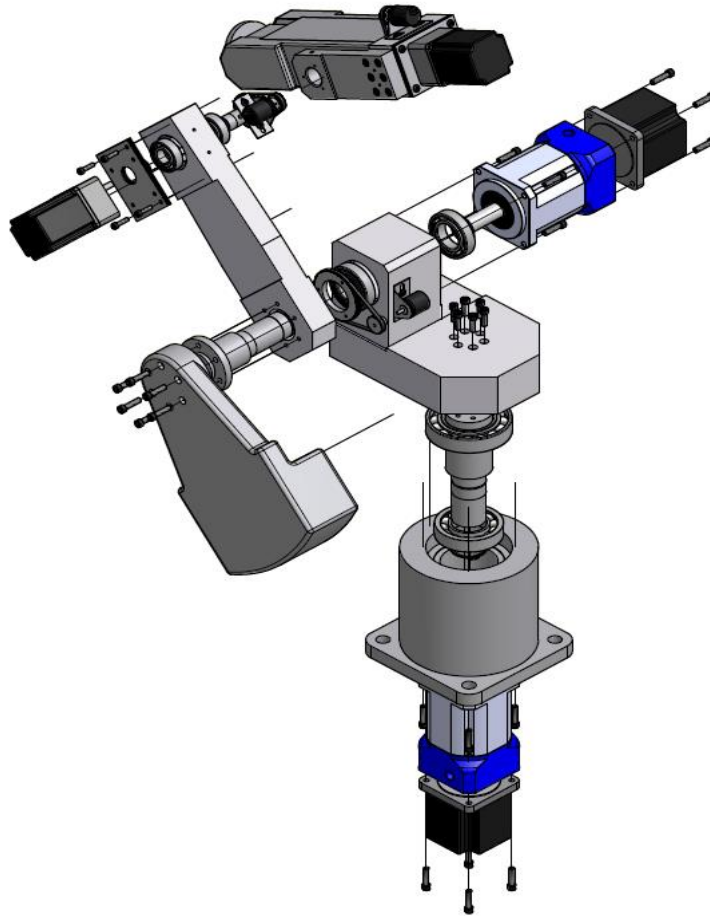


Figura 25 – Vista explodida das três primeiras juntas

Nas juntas 2, 3 e 4 são utilizados dois rolamentos de uma carreira de esfera em cada. Os motores 1 e 2 são acoplados diretamente aos seus redutores, e esses aos eixos das juntas 1 e 2. A Figura 26 mostra os motores 1 e 2 utilizados, e seus redutores de velocidade.



Figura 26 – Motores 1 e 2 utilizados

O Apêndice C apresenta os dados detalhados de cada motor. Todos os motores de passo deste projeto são da empresa Akiyama. O motor 1 é do modelo AK85H/3.75-1.8, torque máximo 52kgf.cm , híbrido, 4 fios, 200 passo por volta e pesa $2,3\text{kgf}$. Os redutores das juntas 1 e 2 são do mesmo modelo, GBPH-0902-NP-020-AA341-500 da Anaheim Automation, com redução de $1:20$, folga de $0,1^\circ$ e peso de $4,2\text{kgf}$. O motor 2 é do modelo AK85H8/3.36-1.8, torque máximo 42kgf.cm , híbrido, 8 fios, 200 passo por volta e pesa $2,3\text{kgf}$.

A Figura 27 mostra os motores 3 e 4. O motor 3 é um motor de passo com redutor de $1:20$ acoplado. O modelo é AK57H/3G20-1.8, torque máximo de 15kgf.cm , híbrido, 6 fios, 200 passo por volta e pesa $1,2\text{kgf}$. O motor 4 é um motor de passo com redutor de $1:10$ acoplado. O modelo é AK57H/3G10-1.8, torque máximo de 15kgf.cm , híbrido, 6 fios, 200 passo por volta e pesa $1,2\text{kgf}$.

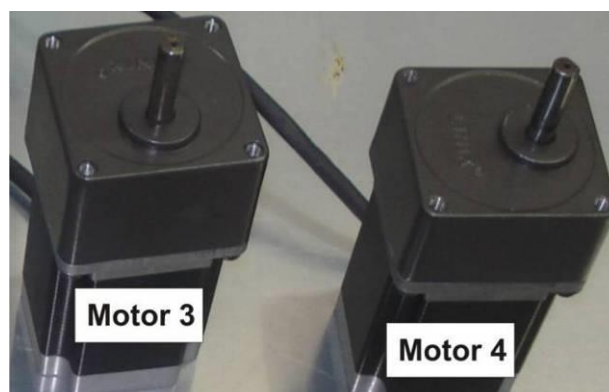


Figura 27 – Motores 3 e 4 utilizados

A Figura 28 mostra os motores 5 e 6, do modelo AK56H/3-1.8, torque máximo de $7,6 \text{ kgf}$, híbrido, 6 fios, 200 passos por volta e com peso $0,6 \text{ kgf}$. Note que os motores mais próximos da extremidade possuem menores pesos, para diminuir o momento de inércia do sistema. Os motores mais próximos à base, por outro lado, precisam executar maiores torques, e portanto são de modelos mais potentes e, conseqüentemente, mais pesados.

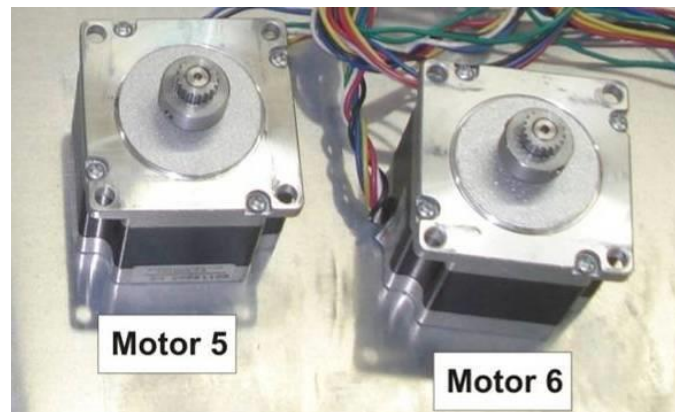


Figura 28 – Foto dos motores 5 e 6

A Figura 29 mostra uma vista explodida das três últimas juntas do manipulador.

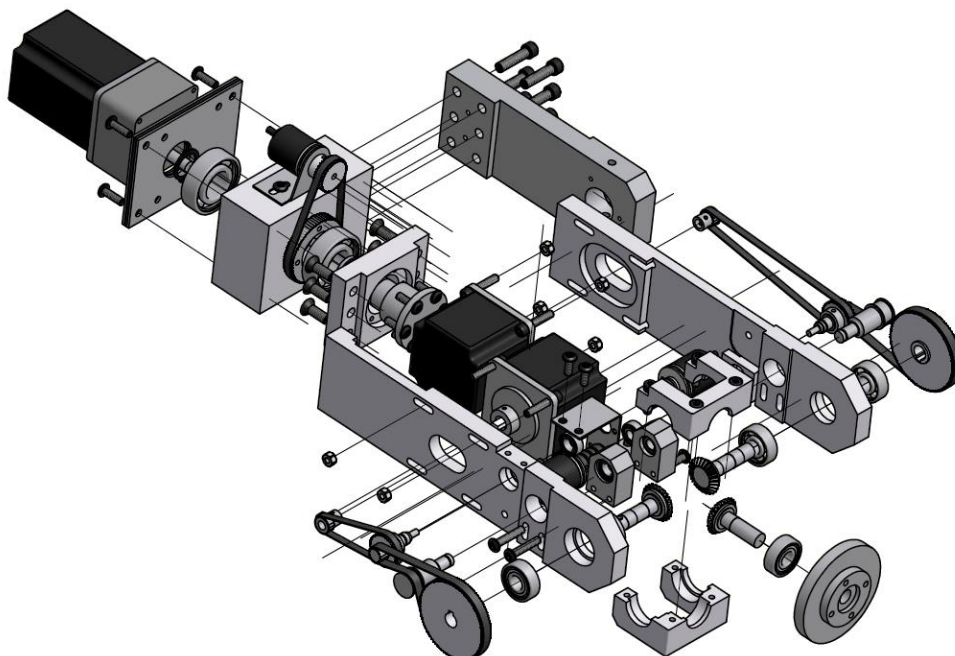


Figura 29 – Vista explodida dos três últimos elos

O motor com redutor da junta 4 é acoplado ao eixo da junta 4. No eixo dos motores das juntas 5 e 6 estão acopladas polias dentadas de 18 dentes. Cada uma

dessas polias aciona uma correia que gira outra polia, de 36 dentes. Esta polia está montada em um eixo com uma polia de 17 dentes na extremidade. Nesta polia está montada uma correia que a liga à outra polia de 85 dentes. Esta polia está montada no eixo de uma das engrenagens cônicas da extremidade. A Figura 30 mostra como esse conjunto é montado.

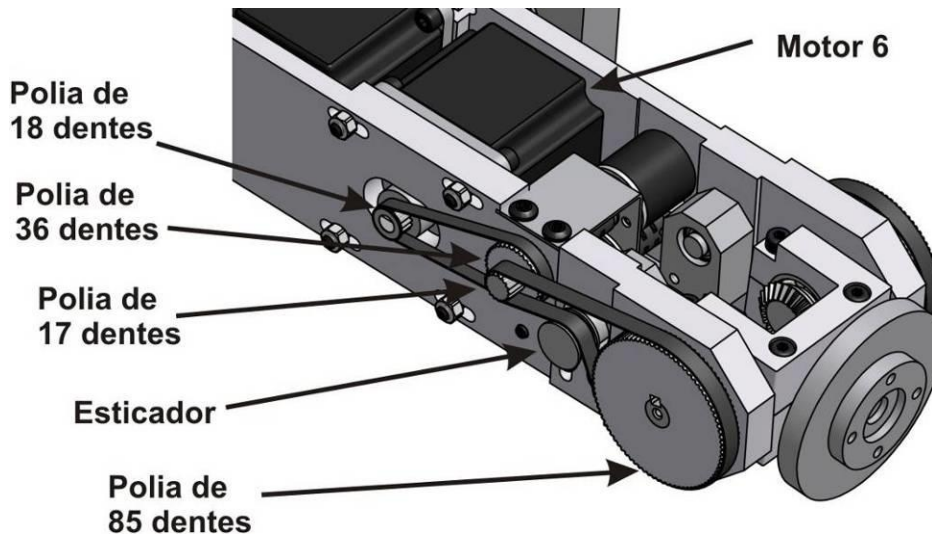


Figura 30 – Detalhe das polias do punho

Assim, pode ser calculada a relação de transmissão entre o motor e o eixo da engrenagem por

$$i_6 = \frac{18}{36} \cdot \frac{17}{85} = \frac{1}{10} \quad (5.2)$$

As juntas 5 e 6 são acionadas pelos motores 5 e 6 simultaneamente. Os motores acionam o conjunto de polias mencionado. Cada polia está acoplada a um eixo de uma engrenagem cônica. A Figura 31 mostra o mecanismo de acionamento da extremidade.

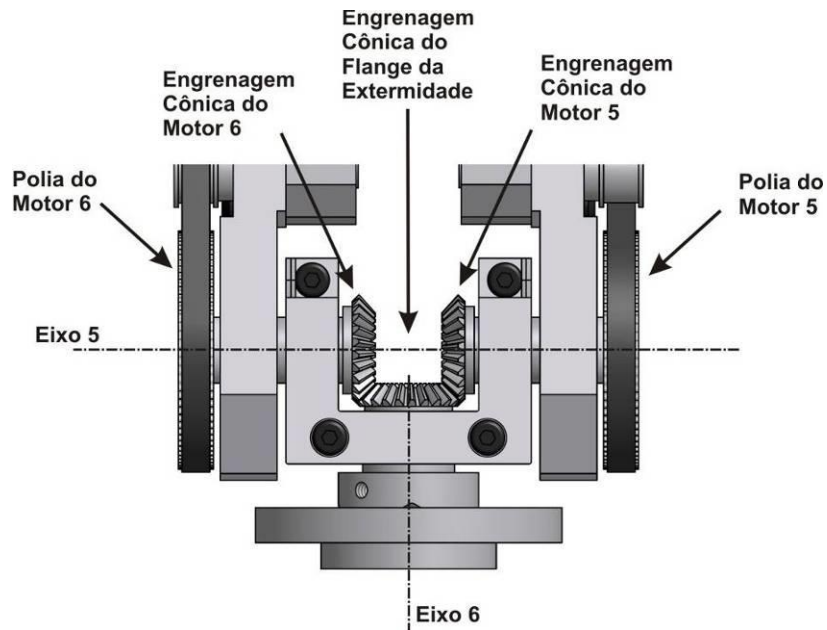


Figura 31 – Detalhe do punho

Quando a engrenagem cônica do motor 6 e a engrenagem cônica do motor 5 giram no mesmo sentido e velocidade, a extremidade do manipulador gira com o mesmo ângulo no eixo da junta 5. Quando as engrenagens são acionadas de forma análoga, mas em sentidos opostos, a extremidade gira ao longo do eixo 6.

Com base nesses dados, a Tabela 3, que relaciona a velocidade dos motores às juntas, é construída.

Tabela 3 – Deslocamento das juntas, onde θ_i é o deslocamento angular

das juntas e δ_i dos motores

Elo	θ_i
1	$\delta_1 / 20$
2	$\delta_2 / 20$
3	$\delta_3 / 20$
4	$\delta_4 / 10$
5	$\delta_5 / 5 + \delta_6 / 5$
6	$\delta_5 / 5 - \delta_6 / 5$

5.1.2. Posição dos centros de gravidade

O projeto do manipulador é feito de modo a reduzir os efeitos da gravidade, e assim maximizar a força útil na extremidade. O peso das juntas 5 e 6 é baixo, logo os torques gravitacionais associados são naturalmente pequenos. A junta 4 gera pouco movimento na direção vertical. Na junta 3, o seu eixo é posicionado de modo a ficar próximo ao centro de gravidade dos elos 4, 5 e 6. A Figura 32 mostra a posição do centro de gravidade desses elos.

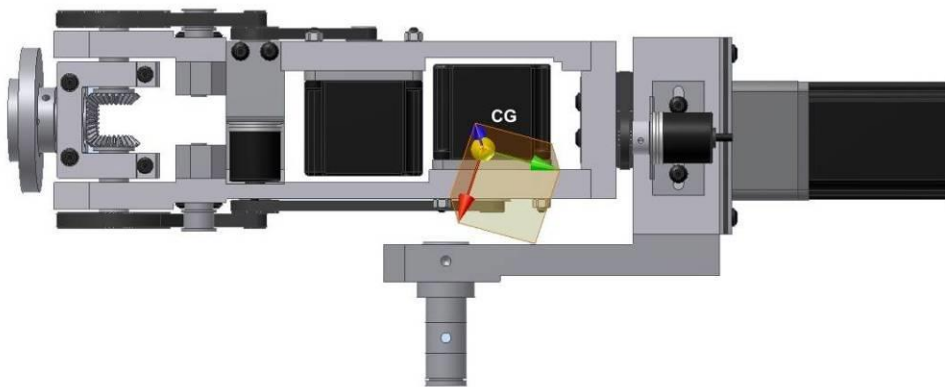


Figura 32 – Posição do centro de gravidade dos três últimos elos

Esta posição do centro de gravidade é devida aos motores 4, 5 e 6 terem sido posicionados o mais próximo possível da parte traseira do elo. Assim, apesar de a extremidade distar até 210mm do eixo 3 quando na posição totalmente distendida, o centro de gravidade fica a apenas 18mm do eixo 3. A posição deste centro de gravidade muda pouco em função do ângulo das juntas 4, 5 e 6.

O centro de gravidade dos elos 2, 3, 4, 5 e 6 é mostrado na Figura 33.

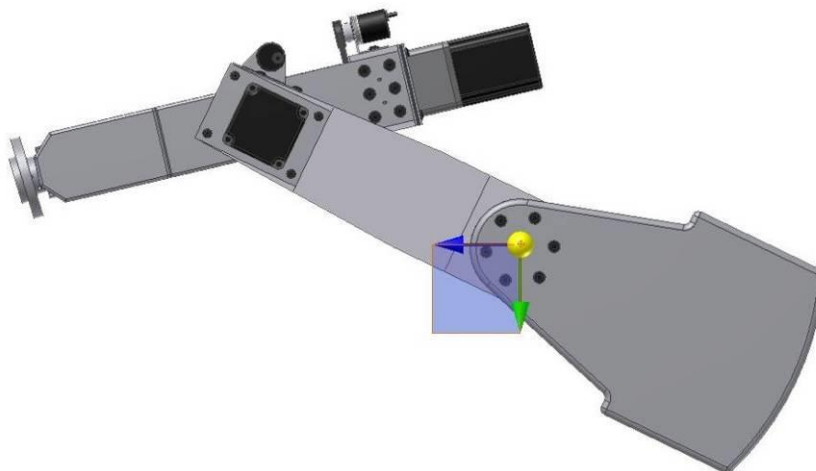


Figura 33 – Posição do centro de gravidade dos quatro últimos elos

Este centro de gravidade fica próximo ao centro do eixo devido ao contra peso montado de forma oposta ao eixo da junta 2. Uma vez que o centro de gravidade dos três últimos elos varia pouco de posição em função da configuração do robô, a posição do centro de gravidade mostrado da Figura 33 também muda muito pouco. A junta 1 gera apenas movimento horizontal, então não sofre influência do torque gravitacional.

5.1.3. Montagem dos sensores de posição

O manipulador possui seis sensores de posição do tipo *encoder* incremental. Todos os *encoders* utilizados são do modelo B-106-23983 da Cytron Technologies. Eles possuem três saídas: duas saídas em quadratura com 500 pulsos por volta cada, e uma de indexação com 1 pulso por volta. A Figura 34 mostra como o *encoder* é montado na junta 1.

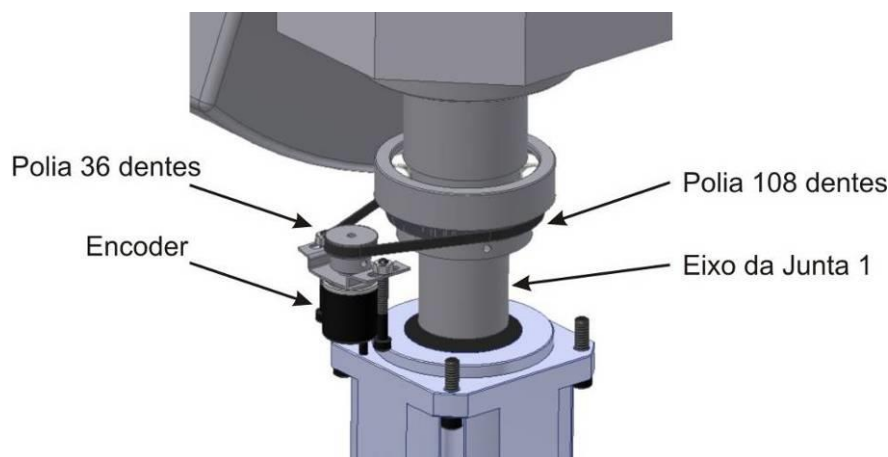


Figura 34 – Montagem do *encoder* na junta 1

Com o controlador configurado para computar o bordo de subida e descida de cada uma das fases, o *encoder* da junta 1 possui resolução de 2000 pulsos por volta. Devido à redução de velocidade proporcionada pelas polias que ligam o *encoder* ao eixo, a cada revolução da junta 1 o sensor identifica 6000 pontos.

A Figura 35 mostra como o *encoder* da junta 2 é montado.

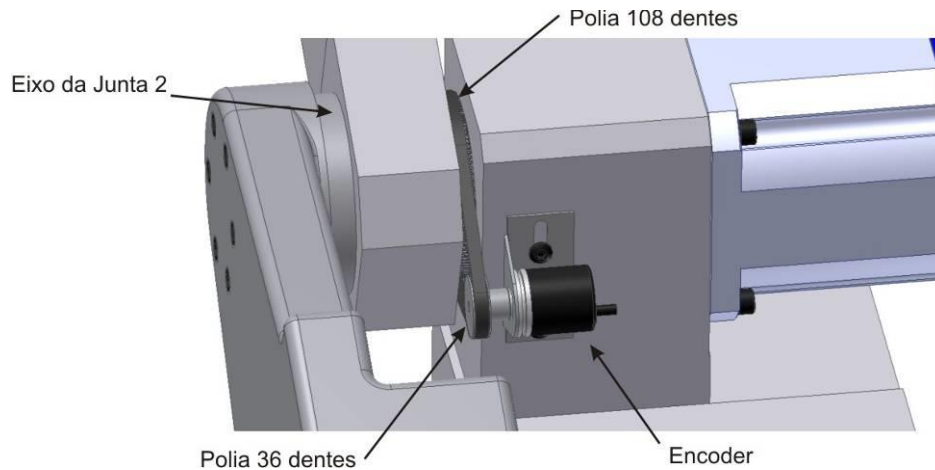


Figura 35 – Montagem do *encoder* na junta 2

Como o controlador desta junta também está configurado para computar o bordo de subida e descida de cada uma das fases, o *encoder* da junta 1 possui resolução de 2000 pulsos por volta. Devido à redução de velocidade proporcionada pelas polias que ligam o *encoder* ao eixo, o sensor da junta 2 também identifica 6000 pontos por volta.

A Figura 36 mostra a montagem dos *encoders* das juntas 3 e 4.

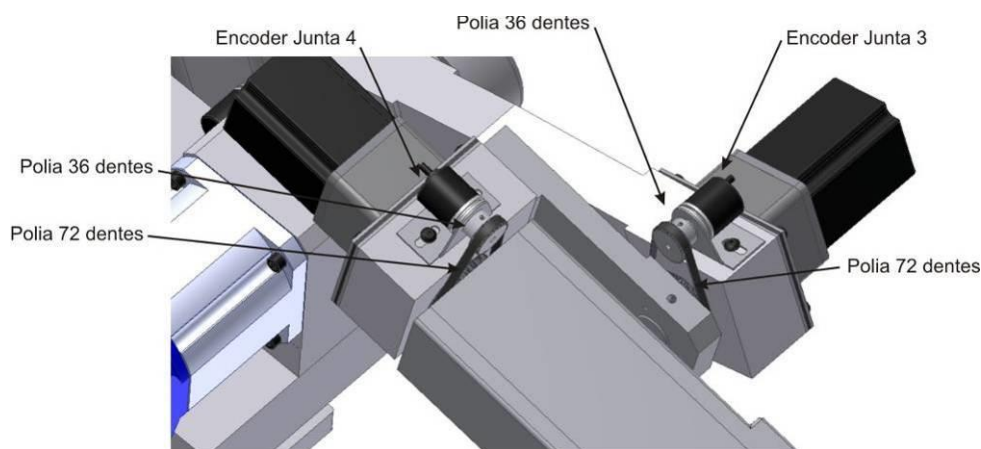


Figura 36 – Montagem dos *encoders* nas junta 3 e 4

Nas juntas 3 e 4, a cada volta das juntas o *encoder* realiza duas voltas. Como os controladores dessas juntas identificam 2000 pulsos por volta do *encoder*, cada junta possui resolução de 4000 pontos por volta, devido às polias.

Nas juntas 5 e 6, os *encoders* estão montados no eixo da polia intermediária, conforme pode ser visto na Figura 37. Como a cada volta do eixo das engrenagens o eixo do *encoder* realiza 5 voltas, e como o controlador identifica 1000 posições por volta do *encoder*, cada eixo lateral do punho possui resolução de 5000 pontos por volta.

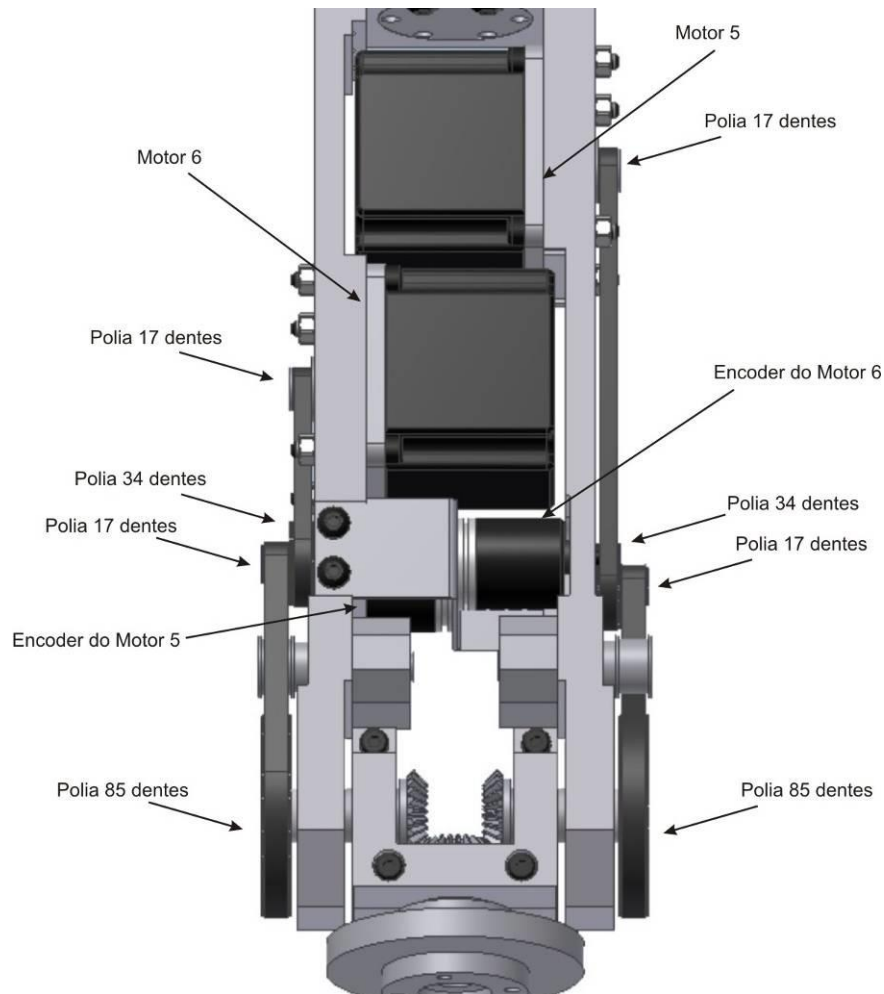


Figura 37 – Montagem dos *encoders* nas juntas 5 e 6

5.2. Projeto do controlador

O projeto dos controladores é feito com base no microcontrolador PIC18F2431. A série de microcontroladores PIC18 possui alta capacidade computacional com baixo custo [26]. Dentro desta série, a família PIC18Fxx31 possui dispositivos para o controle de alto desempenho de motores.

5.2.1. Interface do Encoder

O PIC18F2431 é escolhido por apresentar uma interface para tratamento automático dos sinais do *encoder*, chamada de *Quadrature Encoder Interface* (QEI). O uso desta interface requer uma configuração específica de alguns registradores do microcontrolador para sua inicialização. A Figura 38 apresenta o

fragmento de *datasheet* dos microcontroladores PIC18Fxx31 onde o registrador da QEI é configurado.

REGISTER 16-2: QEICON: QUADRATURE ENCODER INTERFACE CONTROL REGISTER

R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
$\overline{\text{VELM}}$	QERR ⁽¹⁾	UP/DOWN	QEIM2 ^(2,3)	QEIM1 ^(2,3)	QEIM0 ^(2,3)	PDEC1	PDEC0
bit 7						bit 0	

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 7	VELM: Velocity Mode bit 1 = Velocity mode disabled 0 = Velocity mode enabled
bit 6	QERR: QEI Error bit ⁽¹⁾ 1 = Position counter overflow or underflow ⁽⁴⁾ 0 = No overflow or underflow
bit 5	UP/DOWN: Direction of Rotation Status bit 1 = Forward 0 = Reverse
bit 4-2	QEIM2:QEIM0: QEI Mode bits ^(2,3) 111 = Unused 110 = QEI enabled in 4x Update mode; position counter reset on period match (POSCNT = MAXCNT) 101 = QEI enabled in 4x Update mode; INDX resets the position counter 100 = Unused 010 = QEI enabled in 2x Update mode; position counter reset on period match (POSCNT = MAXCNT) 001 = QEI enabled in 2x Update mode; INDX resets the position counter 000 = QEI off
bit 1-0	PDEC1:PDEC0: Velocity Pulse Reduction Ratio bits 11 = 1:64 10 = 1:16 01 = 1:4 00 = 1:1

Figura 38 – Registrador QEI

O *bit 7* determina se o *encoder* está configurado para medir posição ou velocidade. O *bit 6* configura como o contador deve se comportar quando o pulso de *index* do *encoder* é acionado e o contador está com erro. O *bit 5* determina o sentido da contagem. Os *bits 4-2* determinam o modo de operação da QEI. O valor da posição pode ser computado a cada bordo de subida ou descida da fase A, sendo a resolução duas vezes o número de pulsos da fase A. Ou pode ser computado a cada subida e descida dos pulsos de qualquer uma das fases. Assim, a resolução é de quatro vezes o número de pulsos das fases A ou B. Os *bits 0-1* estão relacionados à taxa de medida da velocidade.

A largura mínima do pulso também deve ser informada, em outro registrador, para ser utilizada como filtro. Após essa configuração inicial, a interface QEI atualiza automaticamente o valor da variável posição que está sendo usada no programa. Como neste projeto é necessário interpretar pulsos de entrada

e gerar pulsos de saída, o tratamento automático dos pulsos do *encoder* aumenta o desempenho do sistema.

5.2.2. Visão Geral do Software

A implementação do algoritmo de controle no microcontrolador é feita com o compilador C CCS. Os controladores construídos possuem o mesmo circuito elétrico, mas cada motor necessita de uma programação específica do microcontrolador. A Figura 39 apresenta o circuito elétrico do controlador construído para acionar os motores de passo.

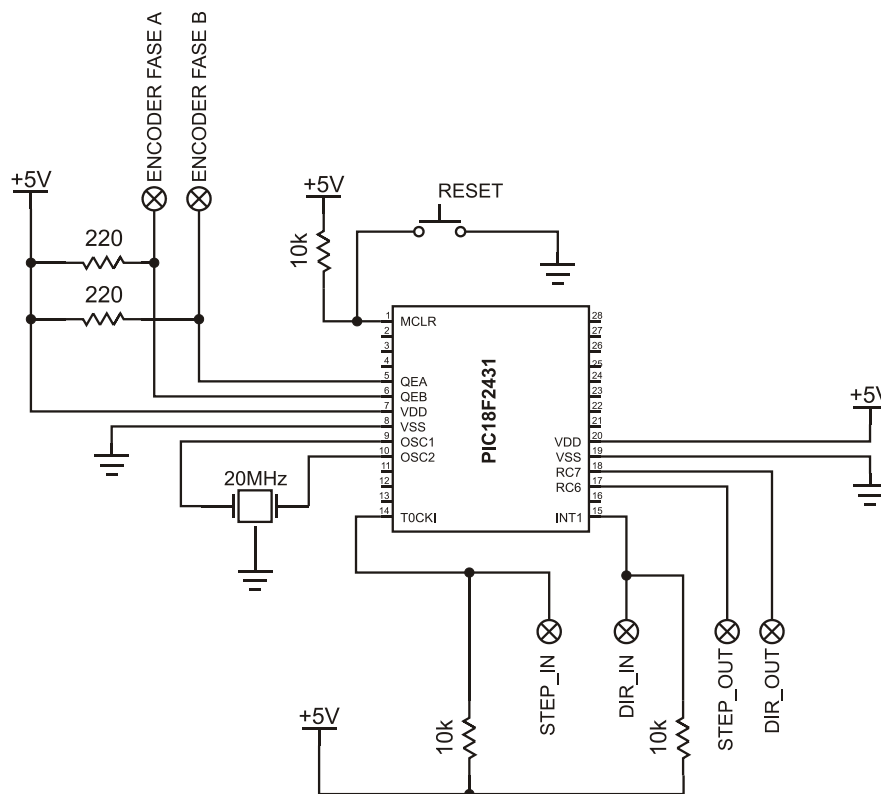


Figura 39 – Diagrama elétrico do controlador

Os pinos QEA e QEB do microcontrolador realizam as leituras das fases do *encoder*. A interface QEI é configurada conforme apresentado na seção anterior. O borne STEP_IN recebe pulsos de passos que o sistema deve realizar, e o borne DIR_IN determina a direção. O pino T0CKI é ligado a STEP_IN. Este pino é configurado como contador de pulsos. No código-fonte, o comando *get_timer0()* retorna os pulsos contados desde o último comando que zera o contador, *set_timer0()*. Durante o *loop* de controle, o sistema lê os pulsos contados. Se a entrada DIR_IN é positiva, esses pulsos contados são somados à posição desejada,

pos_d. Caso contrário, eles são subtraídos. Entretanto, quando o pino DIR_IN muda seu estado, o sistema deve ser capaz de contabilizar os pulsos antes da mudança. Por isso, sempre que a tensão alterna entre nível baixo e alto no pino INT1, uma interrupção é gerada. Essa interrupção, chamada #int_ext1, determina se os pulsos, contados na entrada TOCKI, irão somar ou subtrair a posição desejada. Se a entrada DIR_IN mudar de nível baixo para alto, os pulsos que estavam no contador são contados negativamente. Caso contrário, positivamente. Os pulsos da entrada, STEP_IN, são relativos ao incremento de rotação do *encoder*, e não ao motor. Se o *encoder* possui resolução de 400pulsos/rev. e o motor 200passos/rev., então um incremento unitário da posição demandada ao controlador faz o motor realizar dois passos. O botão RESET reinicia o controlador.

O borne STEP_OUT é a saída de pulsos gerados pelo controlador para o driver do motor, e DIR_OUT o sinal de direção. A frequência dos pulsos de saída está relacionada ao módulo *timer0*. O controlador calcula o valor a ser colocado no contador do *timer0* para que o motor gere a velocidade demandada. A rotina de controle está dentro do *timer0*, mas o período de controle é constante. O controlador calcula quantos ciclos do *timer0* o *loop* de controle deve ser executado para que o período seja de 0,01s. O programa utilizado no microcontrolador está no Apêndice A.

5.3. EMC (Enhanced Machine Controller)

O EMC é um *software* de distribuição livre utilizado para controle em tempo real de tornos, fresadoras, manipuladores seriais e paralelos [27]. Neste trabalho é utilizada a segunda versão deste sistema, o EMC2. Este *software* é modular e flexível, se tornando assim integrável a vários tipos de equipamentos. O EMC começou a ser desenvolvido no NIST (*National Institute of Standards and Technology*, nos EUA) usando sistema operacional Windows NT. O objetivo do NIST era desenvolver um controlador de movimentos para servir como plataforma de teste para conceitos e padrões. Desde o início é solicitado que todo o sistema fosse de domínio público e seu desenvolvimento publicado na Internet. A partir destas publicações, alguns desenvolvedores independentes se interessaram em adaptar o *software* para o sistema operacional Linux, de domínio

público, e arquitetura PC. Este sistema pode ser adaptado em máquinas de controle numérico computadorizado (CNC) em substituição a sistemas originais, muitas vezes antigos e defasados. Após perceberem a contribuição de vários desenvolvedores individuais, o portal LinuxCNC foi criado na internet. Este portal disponibiliza cópias do sistema gratuitamente.

O EMC2 provê várias opções gráficas para a interface homem-máquina. Esta interface aceita a inserção de rotinas usando o código G (segundo padrão RS-274) para programar os movimentos do equipamento. O EMC2 controla até nove atuadores, através de saída PWM com realimentação de posição, ou em malha aberta com saídas de pulso e direção para motores de passo. Alguns dos vários módulos cinemáticos disponíveis são para tornos de dois eixos, fresadoras de até 5 eixos, e manipuladores paralelos e seriais de até seis eixos. A partir da edição de uma série de arquivos de texto, as características da máquina são informadas ao EMC2. Como se trata de um *software* de fonte aberta (*open source*), novos módulos cinemáticos podem ser criados. A Figura 40 mostra um diagrama básico de como os módulos são configurados para que o sistema possa ser integrado ao manipulador utilizado nos testes.

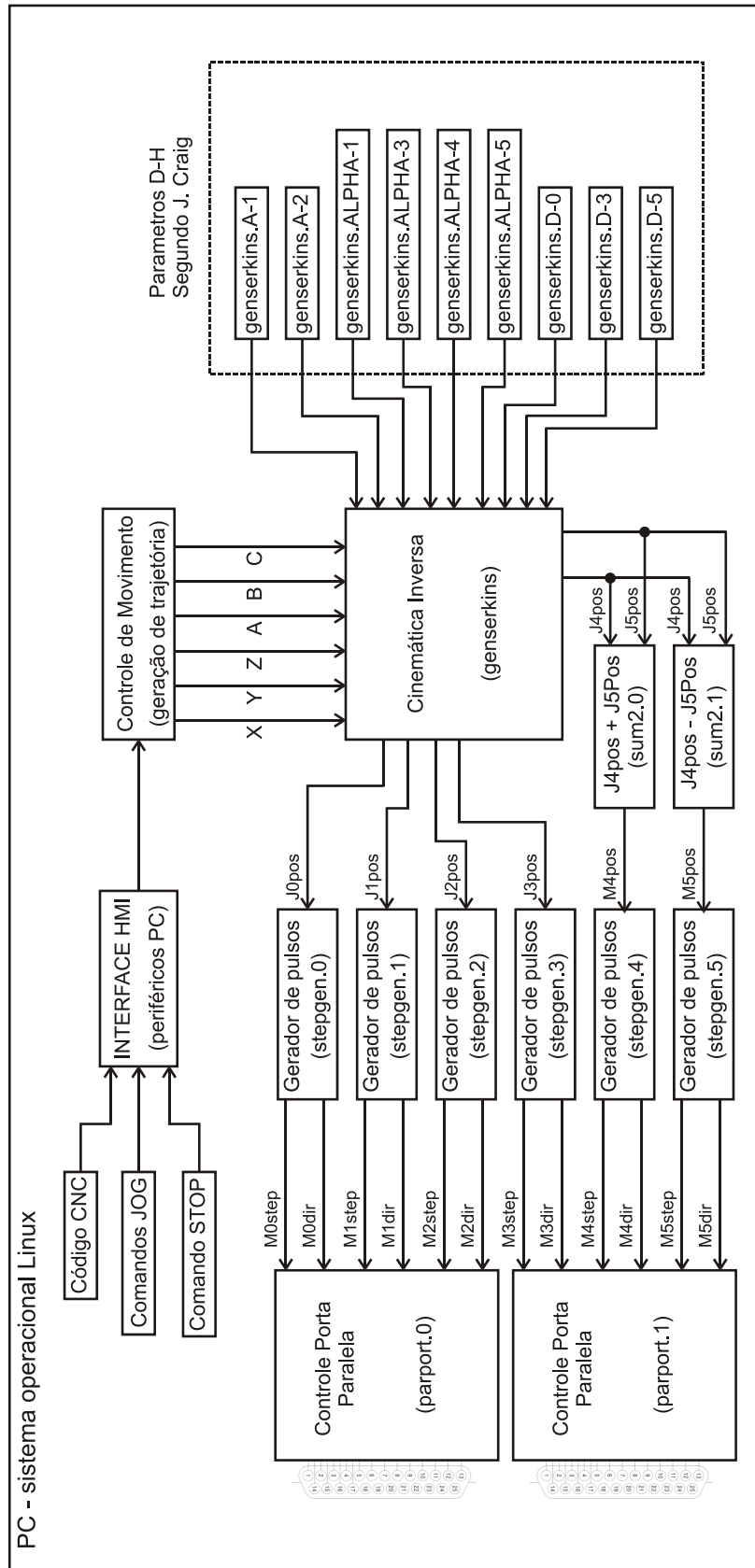


Figura 40 – Diagrama do software implementado

A interface HMI (*Human Machine Interface*) recebe as rotinas de movimento, comandos manuais (JOG) ou comando da parada de emergência (STOP). Esta interface encaminha esses dados para um módulo que gera a trajetória demandada em um espaço cartesiano. O módulo de cinemática inversa (“*genserkins*”) calcula os ângulos a serem colocados em cada junta para alcançar a posição desejada. Este módulo recebe os parâmetros de Denavit-Hartenberg [28]. Os sinais de pulso e direção para o sistema são gerados nos módulos geradores de pulsos (“*stepgen*”). Os ângulos das juntas 1, 2, 3, e 4 são passados diretamente para o módulo gerador de pulsos. Entretanto, as juntas 5 e 6, devido à cinemática do mecanismo do punho, precisam passar por módulos auxiliares para que o ângulo do motor 5 seja a soma dos ângulos das juntas 5 e 6, e para que o ângulo do motor 6 seja a sua diferença. Os pulsos gerados são encaminhados às portas paralelas que estão ligadas aos controladores. A Figura 41 apresenta uma foto do PC com o sistema instalado e o manipulador ao fundo

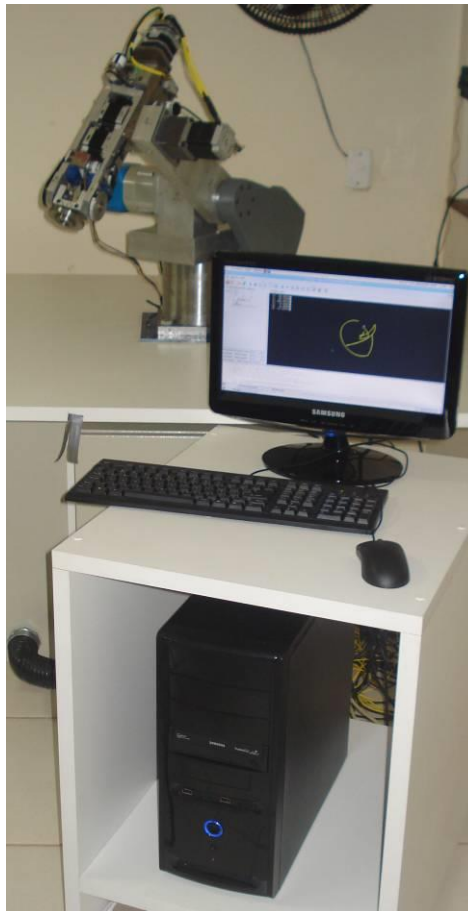


Figura 41 – Foto do computador PC utilizado

A Figura 42 apresenta o diagrama de conexões da junta 1.

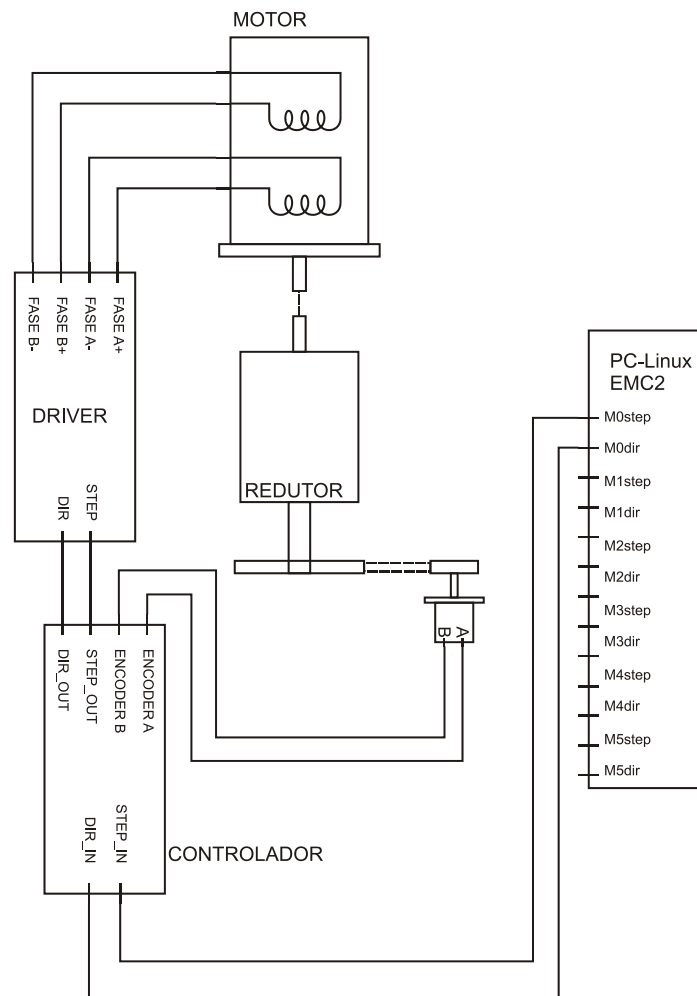


Figura 42 – Diagrama geral de conexões

Os pulsos gerados pelo EMC2 são enviados para o controlador de cada junta. O diagrama apresenta apenas dados referentes à junta 1, mas as demais juntas do manipulador possuem conexões semelhantes. O controlador recebe os pulsos, determina a posição demandada, e executa seu algoritmo de controle. O EMC2 envia os pulsos como se fosse para um sistema em malha aberta. Ou seja, o PC é utilizado para gerar a trajetória demandada em função da entrada feita na interface HMI. Caso o manipulador se desvie muito de sua posição demandada, o controlador deve enviar um sinal de parada de emergência ao EMC2. Entretanto, isto não é implementado no sistema construído.

No próximo capítulo, o manipulador desenvolvido é modelado.

6

Modelagem do manipulador

Neste capítulo, primeiramente a cinemática do manipulador é estudada. A cinemática direta do manipulador é apresentada e, em seguida, a cinemática inversa é calculada.

Após concluir o estudo cinemático do manipulador, é apresentada sua modelagem dinâmica. Nesta análise, os torques que atuam sobre o manipulador, e suas implicações no movimento, são analisados.

6.1.

Cinemática direta

Na realização de qualquer tarefa, a localização da extremidade do manipulador em relação à sua base deve ser conhecida [29]. Na cinemática direta, as variáveis das juntas são assumidas como conhecidas, e o problema se resume em encontrar a posição e orientação da extremidade do manipulador.

A Figura 43 mostra um diagrama esquemático do manipulador construído para esta dissertação, incluindo os sistemas de coordenadas em cada elo seguindo a convenção de Denavit-Hartenberg [29].

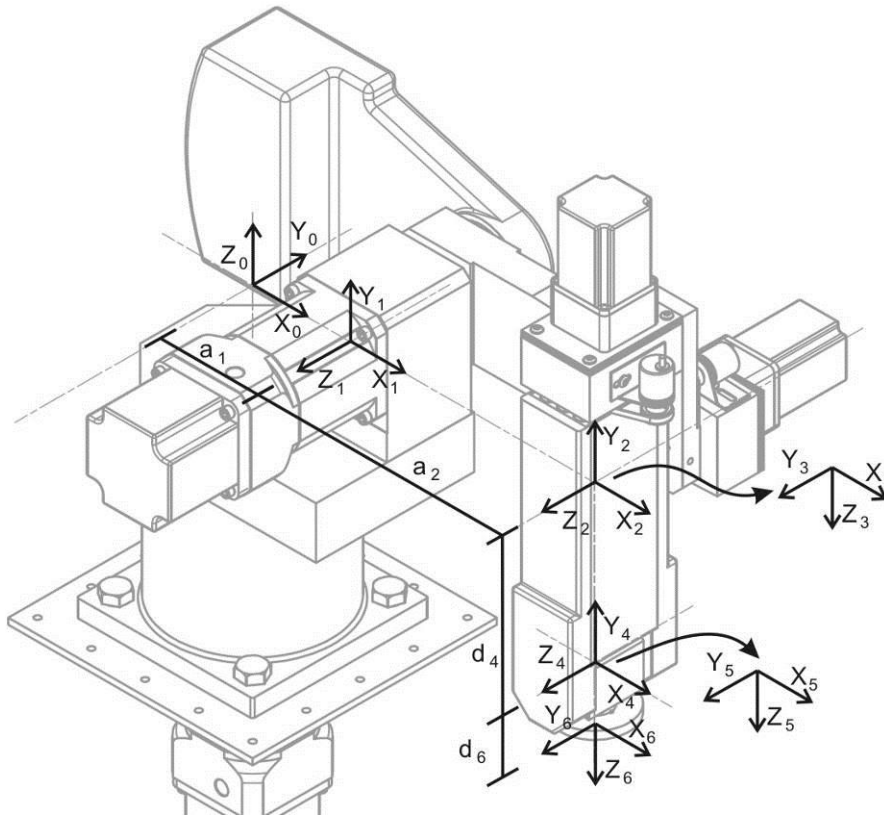


Figura 43 – Diagrama dos eixos coordenados

Usando os sistemas de coordenadas estabelecidos na Figura 43, a Tabela 4 apresenta os parâmetros de Denavit-Hartenberg (DH) de cada elo do manipulador.

Tabela 4 – Parâmetros de DH

Elo	a_i	d_i	α_i	θ_i
1	$a_1 = 0,10m$	0	$+90^\circ$	θ_1
2	$a_2 = 0,25m$	0	0	θ_2
3	0	0	$+90^\circ$	θ_3
4	0	$d_4 = 0,16m$	-90°	θ_4
5	0	0	$+90^\circ$	θ_5
6	0	$d_6 = 0,05m$	0	θ_6

onde:

a_i – comprimento da normal comum entre o eixo da junta i e da junta $i-1$;

d_i – distância entre a origem do sistema $i-1$ e a normal comum;

α_i – ângulo entre o eixo z_{i-1} e o eixo z_i , na direção de x_i ;

θ_i – ângulo entre o eixo x_{i-1} e o eixo x_i , medido na direção de z_{i-1} .

A partir desta tabela, podem-se substituir os valores dos parâmetros de DH para escrever as matrizes homogêneas de transformação, que relacionam as translações e rotações entre os sistemas de coordenadas:

$$A_1^0 = \begin{bmatrix} c_1 & 0 & s_1 & a_1 \cdot c_1 \\ s_1 & 0 & -c_1 & a_1 \cdot s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.1)$$

$$A_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 \cdot c_2 \\ s_2 & c_2 & 0 & a_2 \cdot s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.2)$$

$$A_3^2 = \begin{bmatrix} c_3 & 0 & s_3 & 0 \\ s_3 & 0 & -c_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.3)$$

$$A_4^3 = \begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.4)$$

$$A_5^4 = \begin{bmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.5)$$

$$A_6^5 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.6)$$

onde definem-se $c_i = \cos(\theta_i)$, $s_i = \text{sen}(\theta_i)$, $c_{ij} = \cos(\theta_i + \theta_j)$ e $s_{ij} = \text{sen}(\theta_i + \theta_j)$.

A cinemática direta deste manipulador pode ser obtida através da multiplicação destas matrizes [30] para obter a matriz homogênea que relaciona o sistema da extremidade com o da base

$$A_6^0 = A_1^0 \cdot A_2^1 \cdot A_3^2 \cdot A_4^3 \cdot A_5^4 \cdot A_6^5 \quad (6.7)$$

A localização da extremidade do manipulador pode ser extraída da quarta coluna da matriz A_6^0 de transformação homogênea [29]:

$$A_6^0 = \begin{bmatrix} u_x & v_x & w_x & q_x \\ u_y & v_y & w_y & q_y \\ u_z & v_z & w_z & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.8)$$

onde os elementos q_x , q_y e q_z definem a posição da extremidade em relação ao sistema da base. A orientação da extremidade pode ser obtida através dos cossenos diretores do sistema de coordenadas da extremidade, calculados a partir dos vetores unitários \mathbf{u} , \mathbf{v} e \mathbf{w} nas direções de, \mathbf{x}_6 , \mathbf{y}_6 e \mathbf{z}_6 , respectivamente, vide eq. (6.8).

Após calcular o produto da eq. (6.7) e simplificar as equações usando algumas identidades trigonométricas, os elementos de A_6^0 são obtidos:

$$u_x = c_1 [c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] + s_1(s_4c_5c_6 + c_4s_6) \quad (6.9)$$

$$u_y = s_1 [c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] - c_1(s_4c_5c_6 + c_4s_6) \quad (6.10)$$

$$u_z = s_1(s_4c_5c_6 - s_4s_6) + s_{23}s_5c_6 \quad (6.11)$$

$$v_x = c_1 [-c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6] + s_1(-s_4c_5s_6 + c_4c_6) \quad (6.12)$$

$$v_y = s_1 [-c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6] + s_1(-s_4c_5s_6 + c_4c_6) \quad (6.13)$$

$$v_z = -s_{23}(c_4c_5c_6 + s_4c_6) - c_{23}s_5s_6 \quad (6.14)$$

$$w_x = c_1(c_{23}c_4s_5 + s_{23}c_5) + s_1s_4c_5 \quad (6.15)$$

$$w_y = c_1(c_{23}c_4s_5 + s_{23}c_5) - c_1s_4s_5 \quad (6.16)$$

$$w_z = s_{23}c_4s_5 - c_{23}c_5 \quad (6.17)$$

$$q_x = c_1 [a_1 + a_2c_2 + d_4s_{23} + d_6(c_{23}c_4s_5 + s_{23}c_5)] + d_6s_1s_4s_5 \quad (6.18)$$

$$q_y = s_1 [a_1 + a_2c_2 + d_4s_{23} + d_6(c_{23}c_4s_5 + s_{23}c_5)] - d_6c_1s_4s_5 \quad (6.19)$$

$$q_z = a_2s_2 - d_4c_{23} + d_6(s_{23}c_4s_5 - c_{23}c_5) \quad (6.20)$$

Com esses elementos calculados, a posição e orientação da extremidade a partir dos ângulos das juntas são obtidas. Logo, a cinemática direta do manipulador é conhecida. A partir de alguns elementos da cinemática direta, e utilizando algumas propriedades geométricas do manipulador, a cinemática inversa é calculada a seguir.

6.2. Cinemática inversa

O problema da cinemática inversa é determinar os valores das variáveis das juntas dada uma posição e orientação desejadas da extremidade [30]. Para um manipulador genérico, este problema não possui solução analítica. Para o manipulador desenvolvido, seria necessário resolver o sistema de equações (6.9)-(6.20). No entanto, estas equações são não-lineares e complexas de resolver. Uma solução alternativa, utilizando propriedades geométricas desse manipulador específico, é mais simples de ser obtida, e apresentada a seguir.

A Figura 44 mostra como os sistemas de coordenadas da base e do centro do punho se relacionam. Observando esta figura, a posição do centro do punho (origem do sistema de coordenadas 4) não depende dos ângulos das últimas três juntas. Isso ocorre porque os eixos de rotação das últimas três juntas se interceptam em um único ponto. Assim, a solução da cinemática inversa pode ser dividida em duas partes: obtenção da posição do centro do punho e obtenção da orientação da extremidade.

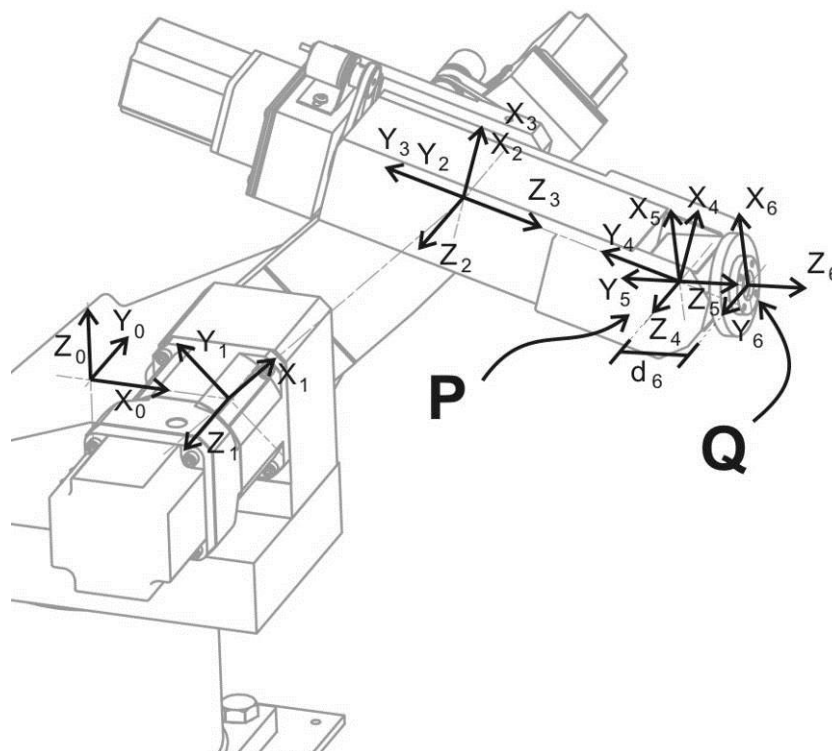


Figura 44 – Posição do centro do punho

Na Figura 44, o sistema 0 é o sistema de coordenadas da base, o ponto P é o centro do punho e o ponto Q é a extremidade do manipulador. A posição do ponto P usando o sistema de coordenadas da extremidade vale

$${}^6p = \overline{QP} = [0 \quad 0 \quad -d_6 \quad 1]^T \quad (6.21)$$

Essa mesma posição no sistema de coordenadas da base vale

$${}^6p = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} q_x - d_6 w_x \\ q_y - d_6 w_y \\ q_z - d_6 w_z \\ 1 \end{bmatrix} \quad (6.22)$$

e no sistema de coordenadas da junta 3 resulta em

$${}^3p = [0 \quad 0 \quad d_4 \quad 1]^T \quad (6.23)$$

Pode-se então calcular 3p usando a matriz de transformação A_3^0 :

$${}^0p = A_3^0 {}^3p \quad (6.24)$$

onde A_3^0 é dada por

$$A_3^0 = A_1^0 A_2^1 A_3^2 = \begin{bmatrix} c_1 c_{23} & s_1 & c_1 & c_1 (a_1 + a_2 c_2) \\ s_1 c_{23} & c_1 & s_1 & s_1 (a_1 + a_2 c_2) \\ s_{23} & 0 & c_{23} & a_2 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.25)$$

Multiplicando-se ambos os lados de (6.24) pela inversa de A_1^0 , obtém-se

$$(A_1^0)^{-1} {}^0p = A_3^1 {}^3p \quad (6.26)$$

e, substituindo as eq.(6.1) – (6.3) em (6.26), resulta em

$$p_x c_1 + p_y s_1 - a_1 = a_2 c_2 + d_4 s_{23} \quad (6.27)$$

$$p_z = a_2 s_2 + d_4 c_{23} \quad (6.28)$$

$$p_x s_1 - p_y c_1 = 0 \quad (6.29)$$

onde p_x , p_y e p_z são dados pela eq. (6.22). A partir da eq. (6.29), uma expressão para θ_1 em função de p_x e p_y pode ser obtida

$$\theta_1 = \tan^{-1} \left(\frac{p_y}{p_x} \right) \quad (6.30)$$

A partir da eq. (6.30), é observado que para cada solução θ_1^* , existe (matematicamente) outra solução $\theta_1^* + \pi$. Mas, devido a limitações mecânicas no

manipulador, somente a primeira solução é válida, a outra não pode ser atingida sem colisão entre os elos.

Elevando os dois lados das eq. (6.27) – (6.29) ao quadrado, e somando todas as equações, obtém-se

$$\kappa_1 = \kappa_2 s_3 \quad (6.31)$$

onde

$$\kappa_1 = p_x^2 + p_y^2 + p_z^2 - 2a_1 p_x c_1 - 2a_1 p_y s_1 + a_1^2 - a_2^2 - d_4^2$$

$$\kappa_2 = 2a_2 d_4$$

A partir da eq. (6.31), a expressão para θ_3 é obtida:

$$\theta_3 = \text{sen}^{-1}\left(\frac{\kappa_1}{\kappa_2}\right) \quad (6.32)$$

Assumindo que $\theta_3 = \theta_3^*$ uma solução, onde $0 \leq \theta_3^* \leq \pi/4$, conclui-se que $\theta_3 = \pi - \theta_3^*$ também é solução. A Figura 45 mostra as duas possibilidades de configuração para a mesma posição do ponto P , conhecidas popularmente em outros manipuladores como “cotovelo para cima” (*upper elbow*) ou “cotovelo para baixo” (*lower elbow*).

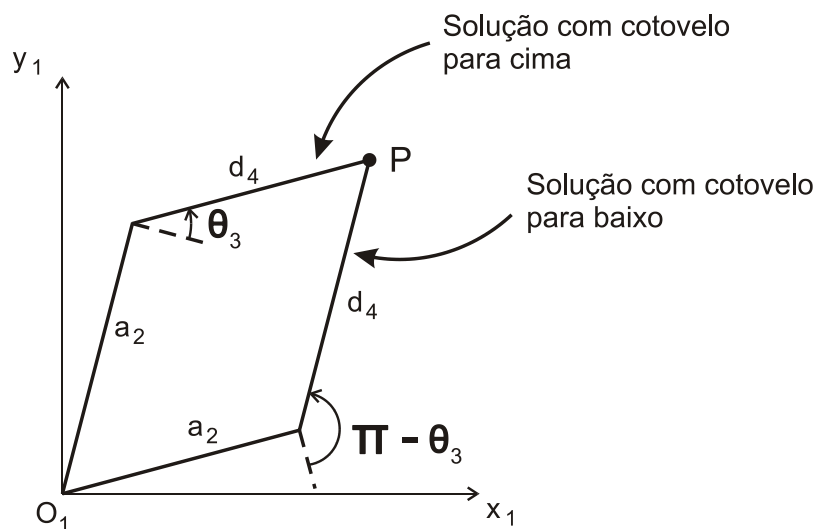


Figura 45 – Duas soluções da junta 3

Devido à possível colisão do motor da junta 4 com os demais elos, somente a solução de θ_3 onde o cotovelo fica para cima é possível. Em seguida, as eq. (6.27) e (6.28) são expandidas, obtendo-se

$$\mu_1 c_2 - \mu_2 s_2 = \gamma_1 \quad (6.33)$$

$$\mu_2 c_2 + \mu_1 s_2 = \gamma_2 \quad (6.34)$$

onde

$$\mu_1 = a_2 + d_4 s_3$$

$$\mu_2 = -d_4 c_3$$

$$\gamma_1 = p_x c_1 + p_y s_1 - a_1$$

$$\gamma_2 = p_z$$

A partir das eq. (6.33) e (6.34) é obtida uma expressão para o seno do ângulo θ_2 da junta 2, s_2 , e uma expressão para o seu cosseno, c_2 :

$$s_2 = \frac{\mu_1 \gamma_2 - \mu_2 \gamma_1}{\mu_1^2 + \mu_2^2} \quad (6.35)$$

$$c_2 = \frac{\mu_1 \gamma_1 + \mu_2 \gamma_2}{\mu_1^2 + \mu_2^2} \quad (6.36)$$

Os valores de s_2 e c_2 são usados para determinar θ_2 por

$$\theta_2 = \text{atan2}(s_2, c_2) \quad (6.37)$$

onde a função $\theta = \text{atan2}(a, b)$ é definida tal que $\sin\theta = a$ e $\cos\theta = b$.

A partir das equações apresentadas, duas possíveis soluções para a junta 1 e duas para junta 3 são obtidas, o que resulta em quatro possíveis combinações. Entretanto, devido a limitações mecânicas do manipulador anteriormente mencionadas, apenas uma solução da cinemática inversa é possível de ser alcançada.

Uma vez que se conhece a posição do centro do punho do manipulador em relação à sua base, é necessário cálculo da orientação da extremidade para obter a solução completa da cinemática inversa. Como θ_1 , θ_2 e θ_3 já foram calculados pelas equações anteriores, a matriz homogênea A_3^0 relacionando os sistemas 0 e 3 é conhecida. Esta matriz satisfaz a relação

$$A_6^0 = A_3^0 A_6^3 \quad (6.38)$$

Multiplicado ambos os lados da eq. (6.38) pela inversa de A_3^0 , obtém-se

$$A_6^3 = (A_3^0)^{-1} A_6^0 \quad (6.39)$$

Como o elemento 3x3 de A_6^3 é igual ao cosseno do ângulo da junta 5, c_5 , parte da multiplicação do lado direito da eq. (6.39) pode ser efetuada para calcular este elemento, como representado a seguir, onde os elementos que não participam da multiplicação estão suprimidos por “_”:

$$\begin{bmatrix} - & - & - & - \\ - & - & - & - \\ - & - & c_5 & - \\ - & - & - & - \end{bmatrix} = \begin{bmatrix} - & - & - & - \\ - & - & - & - \\ c_1 s_{23} & s_1 s_{23} & -c_{23} & - \\ - & - & - & - \end{bmatrix} \begin{bmatrix} - & - & w_x & - \\ - & - & w_y & - \\ - & - & w_z & - \\ - & - & 0 & - \end{bmatrix} \quad (6.40)$$

A partir desta multiplicação matricial, é possível obter uma expressão para θ_5 :

$$\theta_5 = \cos^{-1}(w_x c_1 s_{23} + s_1 s_{23} w_y - c_{23} w_z) \quad (6.41)$$

Como o elemento 1x3 de A_6^3 é igual a $c_4 s_5$, ele pode ser calculado a partir da eq. (6.39), como representado a seguir:

$$\begin{bmatrix} - & - & c_4 s_5 & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix} = \begin{bmatrix} c_1 s_{23} & s_1 s_{23} & s_{23} & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix} \begin{bmatrix} - & - & w_x & - \\ - & - & w_y & - \\ - & - & w_z & - \\ - & - & 0 & - \end{bmatrix} \quad (6.42)$$

A partir desta multiplicação matricial, obtém-se uma expressão para o cosseno do ângulo da junta 4, c_4 , válida desde que $s_5 \neq 0$:

$$c_4 = \frac{w_x c_1 c_{23} + w_y s_1 c_{23} + w_z s_{23}}{s_5} \quad (6.43)$$

A matriz A_6^3 possui o elemento 2x3 igual a $s_4 s_5$, portanto a partir da eq. (6.39) pode-se escrever

$$\begin{bmatrix} - & - & - & - \\ - & - & s_4 s_5 & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix} = \begin{bmatrix} - & - & - & - \\ s_1 & -c_1 & 0 & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix} \begin{bmatrix} - & - & w_x & - \\ - & - & w_y & - \\ - & - & w_z & - \\ - & - & 0 & - \end{bmatrix} \quad (6.44)$$

A partir desta multiplicação matricial, obtém-se uma expressão para s_4 , válida desde que $s_5 \neq 0$:

$$s_4 = \frac{w_x s_1 - w_y c_1}{s_5} \quad (6.45)$$

A partir dos valores de s_4 e c_4 calculados, pode-se calcular o valor de θ_4 :

$$\theta_4 = \text{atan2}(s_4, c_4) \quad (6.46)$$

Como o elemento 3x1 de A_6^3 é igual a $-s_5 c_6$, parte da multiplicação do lado direito da eq. (6.39) resulta neste elemento:

$$\begin{bmatrix} - & - & - & - \\ - & - & - & - \\ -s_5 c_6 & - & - & - \\ - & - & - & - \end{bmatrix} = \begin{bmatrix} - & - & - & - \\ - & - & - & - \\ c_1 s_{23} & s_1 s_{23} & -c_{23} & - \\ - & - & - & - \end{bmatrix} \begin{bmatrix} u_x & - & - & - \\ u_y & - & - & - \\ u_z & - & - & - \\ 0 & - & - & - \end{bmatrix} \quad (6.47)$$

e, a partir desta multiplicação matricial, obtém-se c_6 :

$$c_6 = -\frac{u_x c_1 s_{23} + u_y s_1 s_{23} - u_z c_{23}}{s_5} \quad (6.48)$$

Analogamente, como o elemento 3x2 de A_6^3 é igual a $s_5 s_6$, pode-se escrever

$$\begin{bmatrix} - & - & - & - \\ - & - & - & - \\ - & s_5 s_6 & - & - \\ - & - & - & - \end{bmatrix} = \begin{bmatrix} - & - & - & - \\ - & - & - & - \\ c_1 s_{23} & s_1 s_{23} & -c_{23} & - \\ - & - & - & - \end{bmatrix} \begin{bmatrix} v_x & - & - & - \\ v_y & - & - & - \\ v_z & - & - & - \\ 0 & - & - & - \end{bmatrix} \quad (6.49)$$

E, a partir desta multiplicação matricial, obtém-se s_6 :

$$s_6 = \frac{v_x c_1 s_{23} + v_y s_1 s_{23} - v_z c_{23}}{s_5} \quad (6.50)$$

Finalmente, a partir dos valores de s_6 e c_6 calculados, obtém-se o valor do ângulo da junta da 6, θ_6 :

$$\theta_6 = \text{atan2}(s_6, c_6) \quad (6.51)$$

Devido limitações mecânicas da junta 5, seu espaço de trabalho está compreendido entre, aproximadamente $\pm 90^\circ$. Logo, quando $s_5 = 0$, conclui-se que $\theta_5 = 0$. Nesta situação singular, a solução da eq. (6.46) para θ_4 não pode ser usada. Quando isto ocorre, apenas a diferença entre θ_4 e θ_6 pode ser computada, porque os eixos z_4 e z_6 se tornam coincidentes.

Uma vez conhecido o comportamento cinemático do manipulador, pode-se derivar seu comportamento dinâmico, como descrito a seguir.

6.3. Dinâmica

O desenvolvimento do modelo dinâmico do manipulador é importante para diversos fins. Primeiro, um modelo dinâmico pode ser usado para simular o comportamento do manipulador sobre várias condições de operação. Segundo, pode ser usado para desenvolver estratégias de controle mais eficientes. Terceiro, a análise dinâmica do manipulador permite calcular todas as forças e torques necessários para seguir trajetórias de interesse, e assim poder dimensionar os seus elos, atuadores e mancais [29].

A equação de movimento para um manipulador serial genérico é descrita por [28]

$$\sum_{j=0}^n M_{ij} \ddot{q}_j + V_i + G_i = Q_i \quad \text{para } i = 1, 2, \dots, n. \quad (6.52)$$

onde:

$$V_i = \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial M_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial M_{jk}}{\partial q_i} \right) \dot{q}_j \dot{q}_k \quad (6.53)$$

$$G_i = - \sum_{j=1}^n m_j \bar{g}^{-T} J_{vj}^i \quad (6.54)$$

Na eq. (6.52), a matriz M é a matriz de inércia do manipulador, q é o vetor de coordenadas generalizadas das juntas, n é o número de graus de liberdade, V é o vetor com termos associados a forças inerciais centrífugas e de Coriolis, G_i é o torque devido à força da gravidade atuando na junta i , Q_i é a força generalizada correspondente a junta i , J_{vj}^i é a j -ésima coluna da matriz Jacobiana linear do elo i , m_j é a massa do elo j , e \bar{g} é o vetor da aceleração gravitacional.

A matriz M pode ser calculada a partir de

$$M = \sum_{i=1}^n \left(J_{vi}^T m_i J_{vi} + J_{oi}^T I_i J_{oi} \right) \quad (6.55)$$

onde J_{oi} é a sub-matriz Jacobiana associada à velocidade angular do elo i , J_{vi} é a sub-matriz Jacobiana associada à velocidade linear do centro de massa do elo i , e I_i é a matriz de inércia do elo i em relação ao centro de massa, usando o sistema de coordenadas da base [28].

As massas de cada elo, obtidas a partir do *software* CAD utilizado no projeto, são

$$m_1 = 12,1kg \quad (6.56)$$

$$m_2 = 21,7kg \quad (6.57)$$

$$m_3 = 3,69kg \quad (6.58)$$

$$m_4 = 4,02kg \quad (6.59)$$

$$m_5 = 0.203kg \quad (6.60)$$

$$m_6 = 0.369kg \quad (6.61)$$

e as matrizes de inércia dos elos, em relação a seus respectivos centros de gravidade e usando os sistemas de coordenadas locais de cada junta, são

$${}^1I_1 = \begin{bmatrix} 0,1314 & -0,0178 & 0,0143 \\ -0,0178 & 0,1314 & -0,0250 \\ 0,0143 & -0,0250 & 0,0727 \end{bmatrix} kg.m^2 \quad (6.62)$$

$${}^2I_2 = \begin{bmatrix} 0,0626 & 0,0028 & -0,0050 \\ -0,0028 & 0,5190 & -0,0002 \\ 0,0050 & -0,0002 & 0,5418 \end{bmatrix} kg.m^2 \quad (6.63)$$

$${}^3I_3 = \begin{bmatrix} 0,0222 & 0,0002 & 0 \\ 0,0002 & 0,0196 & 0,0058 \\ 0 & 0,0058 & 0,0059 \end{bmatrix} kg.m^2 \quad (6.64)$$

$${}^4I_4 = \begin{bmatrix} 0,0281 & 0 & 0 \\ 0 & 0,0044 & 0 \\ 0 & 0 & 0,0260 \end{bmatrix} kg.m^2 \quad (6.65)$$

$${}^5I_5 = \begin{bmatrix} 0,00010 & 0 & 0 \\ 0 & 0,00012 & 0 \\ 0 & 0 & 0,00011 \end{bmatrix} kg.m^2 \quad (6.66)$$

$${}^6I_6 = \begin{bmatrix} 0,00011 & 0 & 0 \\ 0 & 0,00011 & 0 \\ 0 & 0 & 0,00017 \end{bmatrix} kg.m^2 \quad (6.67)$$

Os vetores que descrevem a posição do centro de gravidade do elo i , usando o sistema de coordenadas local do próprio elo i , são

$${}^1r_{c1} = \begin{bmatrix} -0,027228 \\ -0,047041 \\ 0,041620 \end{bmatrix} m \quad (6.68)$$

$${}^2r_{c2} = \begin{bmatrix} -0,338365 \\ 0,000378 \\ -0,132453 \end{bmatrix} m \quad (6.69)$$

$${}^3r_{c3} = \begin{bmatrix} 0,001933 \\ -0,031652 \\ -0,113306 \end{bmatrix} m \quad (6.70)$$

$${}^4r_{c4} = \begin{bmatrix} -0,000103 \\ 0,119374 \\ 0,000103 \end{bmatrix} m \quad (6.71)$$

$${}^5r_{c5} = \begin{bmatrix} 0 \\ 0,000257 \\ 0,004616 \end{bmatrix} m \quad (6.72)$$

$${}^6r_{c6} = \begin{bmatrix} 0 \\ 0 \\ 0,012196 \end{bmatrix} m \quad (6.73)$$

Como este manipulador possui apenas juntas rotativas, a matriz J_{vi} pode ser calculada a partir de

$$J_{vi}^j = z_{j-1} \times {}^{j-1}p_{ci}^* \quad (6.74)$$

onde ${}^{j-1}p_{ci}^*$ é o vetor que define a posição do centro de massa do elo i , em relação à origem do elo $j-1$, escrito no sistema de coordenadas da base. Este vetor pode ser calculado usando as eqs. (6.68-6.73) e as eqs. (6.1-6.6). A coluna j é definida como nula se $j > i$, uma vez que a velocidade do centro de massa de um elo i de um manipulador serial não depende dos ângulos das juntas posteriores a ele, entre $i+1$ e n .

O cálculo das forças centrífugas e de Coriolis necessitam das derivadas parciais da matriz de inércia em relação às coordenadas das juntas. Devido à quantidade de termos da matriz de inércia, aproximações numéricas das derivadas são calculadas. As aproximações das forças centrífugas e de Coriolis são calculadas por

$$V_i = \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\Delta M_{ij}}{\Delta q_k} - \frac{1}{2} \frac{\Delta M_{jk}}{\Delta q_i} \right) \dot{q}_j \dot{q}_k \quad (6.75)$$

onde Δq_k e Δq_i são incrementos pequenos nas coordenadas das juntas k e i , respectivamente. A aproximação da derivada com relação a q_k é calculada a partir da diferença entre a matriz de inércia calculada com $q_k + \Delta q_k$ e a matriz original, e essa diferença é dividida por Δq_k . Esta derivada é estimada por

$$\frac{\Delta M_{ij}}{\Delta q_k} = \frac{M_{ij}(q_k + \Delta q_k) - M_{ij}(q_k)}{\Delta q_k} \quad (6.76)$$

Analogamente, a derivada com relação a q_i é estimada por

$$\frac{\Delta M_{jk}}{\Delta q_i} = \frac{M_{jk}(q_i + \Delta q_i) - M_{jk}(q_i)}{\Delta q_i} \quad (6.77)$$

Desse modo, o estudo da dinâmica do manipulador está completo. Com a cinemática direta e inversa calculadas, o modelo do manipulador proposto está concluído, e pode ser simulado, como descrito no próximo capítulo.

7 Simulações

Neste capítulo, é apresentada primeiramente a simulação do controle de um motor de passo. Em seguida, o controle de seis motores aplicados ao manipulador proposto é estudado. As simulações são feitas usando o *software Simulink's SimPowerSystem*.

7.1. Modelo de motor de passo

O modelo de motor de passo, apresentado no capítulo 3, está disponível na biblioteca de blocos do Simulink. Este bloco permite que motores de relutância variável e motores de ímã permanente (ou híbridos) sejam simulados. O modelo de motor consiste de uma parte elétrica e uma parte mecânica [21]. A parte elétrica é baseada no circuito equivalente composto com um resistor e um indutor em série. Este circuito representa bem o motor se o circuito magnético não possui saturação e se a indutância mútua entre as fases é desprezível. A parte mecânica do motor é baseada em um rotor com inércia e atrito viscoso. A Figura 46 mostra o ícone do modelo de motor de passo do Simulink.

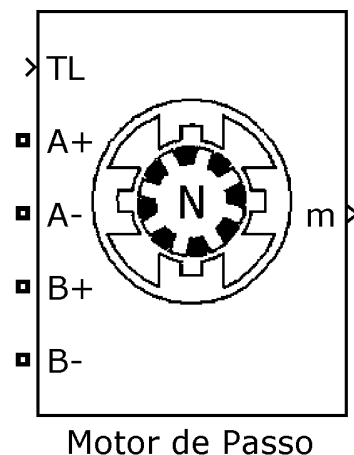


Figura 46 – Ícone do motor de passo do Simulink

O terminal TL recebe o torque da carga. Os terminais $A+$, $A-$, $B+$ e $B-$ se conectam ao *driver* para serem alimentados. A saída m é um vetor contendo cinco sinais: tensão das fases, corrente das fases, torque eletromagnético, velocidade do

rotor e posição do rotor. A Figura 47 apresenta a caixa de diálogo de configuração deste bloco.

Block Parameters: Motor de Passo

Stepper Motor (mask) (link)

Depending on the motor configuration specified by the Motor type parameter, this block models:

- a two or four phases permanent-magnet / hybrid stepper motor,
- a three, four, or five phases variable reluctance stepper motor.

Parameters

Motor type: Permanent-magnet / Hybrid

Number of phases: 2

Winding Inductance (H): 10e-3

Winding resistance (Ohm): 1.2

Step angle (degree): 1.8

Maximum Flux Linkage (Vs): 0.04

Maximum Detent Torque (N.m): 0.02

Total inertia (kg.m.m): 0.25e-5+0.0001*0

Total friction (kg.m/s): 1e-3+50e-3*1

Initial speed (rad/s): 0

Initial position (degrees): 0

Sample time (-1 for inherited): -1

Buttons: OK, Cancel, Help, Apply

Figura 47 – Caixa de diálogo do motor de passo do Simulink

Segue abaixo a descrição de cada parâmetro solicitado:

Motor type – Tipo de motor simulado: ímã permanente/híbrido ou relutância variável.

Number of phases – Número de fases: 2 ou 4 fases.

Winding inductance – Indutância em Henry (H) de cada fase.

Winding resistance – Resistência em ohm (Ω) de cada fase.

Step angle – Ângulo de passo em graus.

Maximum flux linkage – Máximo fluxo induzido nas fases ($V.s$).

Maximum detent torque – Torque máximo de retenção devido ao imã permanente ($N.m$).

Total inertia – Inércia do rotor mais a carga ($kg.m^2$).

Total friction – Coeficiente de atrito total, motor mais carga, ($N.m.s$).

Initial speed - Velocidade angular inicial (rad/s).

Initial position – Posição inicial do rotor ($graus$).

A seguir é apresentado o modelo do driver que aciona este modelo de motor.

7.2.

Modelo de driver

O *driver* alimenta os terminais do motor através de um conversor bipolar a partir de duas fontes de tensão controladas por um sequenciador de fases. A Figura 51 apresenta o modelo de *driver* implementado.

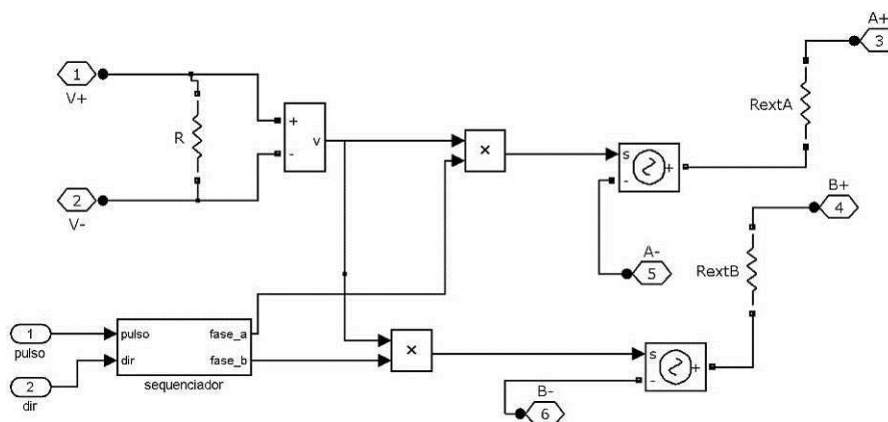


Figura 48 – Modelo de *driver* no Simulink

As saídas do sequenciador, *fase_a* e *fase_b*, podem assumir três valores: -1 , 0 e $+1$. Esses valores são multiplicados pela tensão entre os terminais $V+$ e $V-$, e o valor desse produto é colocado em um gerador de tensão equivalente a esses sinais.

O sequenciador recebe duas entradas, pulso e direção, e determina como as fases serão excitadas. O terminal direção determina o sentido da rotação, enquanto o terminal pulso determina que um incremento na rotação deve ser dado. Este

modelo de sequenciador pode ser utilizado em modo meio passo ou passo completo. A Figura 52 mostra esquematicamente como, em função dos sinais de entrada, as fases são energizadas em modo meio passo. Este gráfico é coerente com a Tabela 2, apresentada anteriormente. O modo passo completo é obtido se a sequência de energização seguir a Tabela 1. Os dois primeiros gráficos apresentam as entradas do sequenciador. Os dois últimos mostram as saídas do *driver* para os terminais do motor de passo.

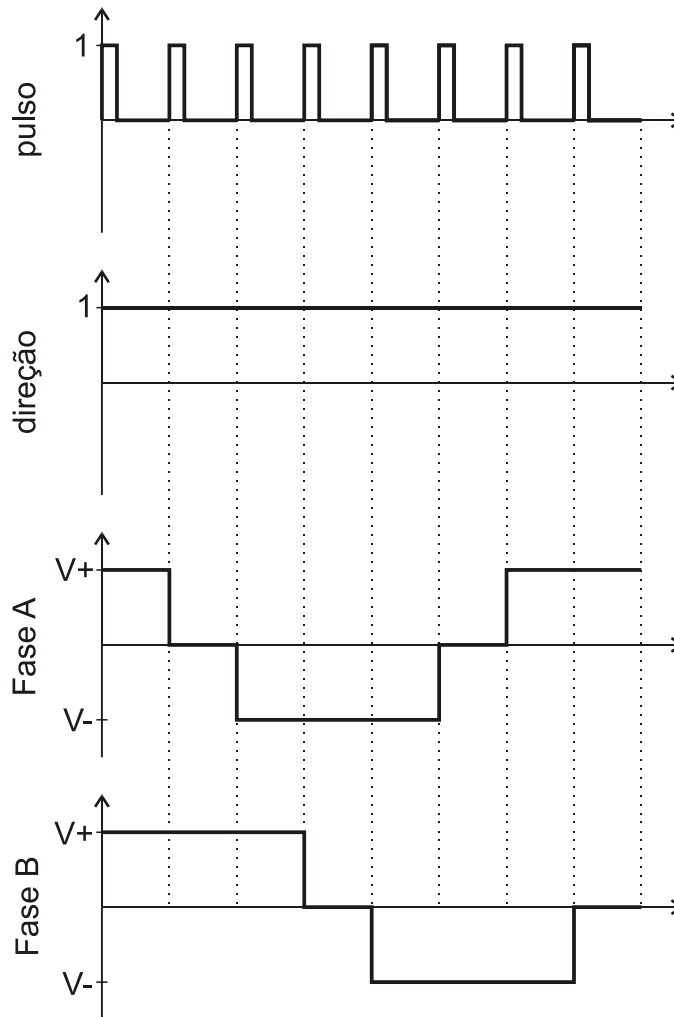


Figura 49 – Entrada e saída do sequenciador

Na Figura 52, o sequenciador recebe nove pulsos de passo, um a cada $0,01s$, sempre na mesma direção. Se o motor utilizado for de 200 passos por volta, uma rotação de $0,9^\circ$ deve ser realizada a cada pulso. A entrada de direção é constante, logo o motor gira em um só sentido. A seguir é apresentado o gerado de pulsos que envia os pulsos para o sequenciador.

7.3. Gerador de pulsos

Segundo a malha de controle proposta no Capítulo 4, os pulsos para o motor de passo devem ser gerados em função de uma velocidade demandada. Ou seja, o gerador de pulsos deve gerar pulsos de passo e direção que gerem uma velocidade no motor proporcional ao sinal de entrada. A Figura 53 apresenta o modelo de gerador implementado no Simulink.

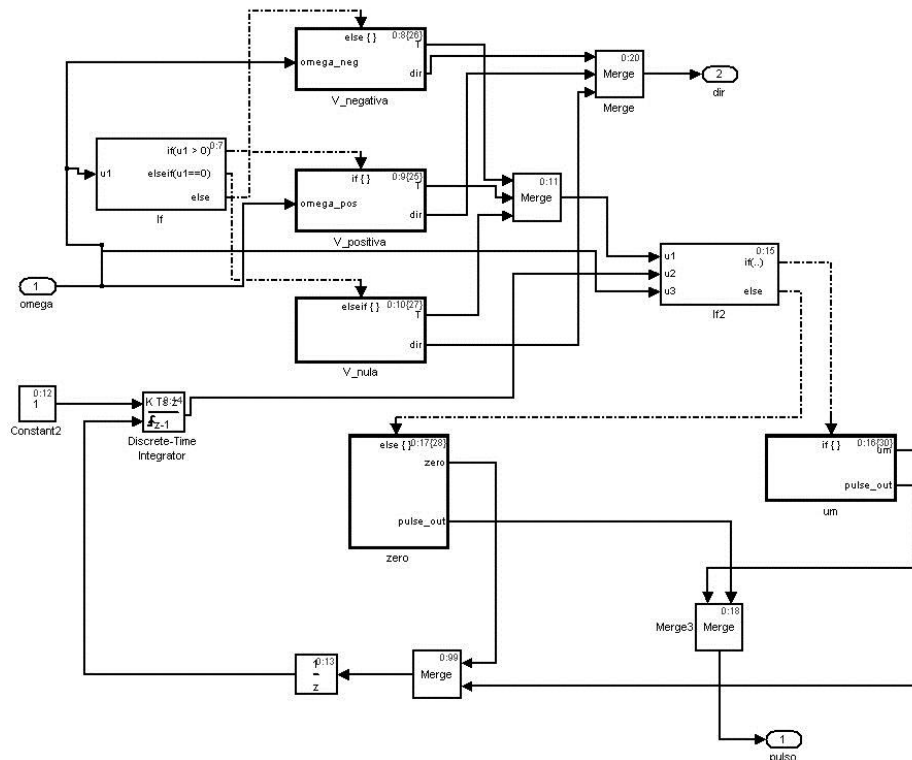


Figura 50 – Modelo de gerador de pulsos

Este gerador de pulsos recebe a velocidade através do terminal *omega* e, se for positiva, coloca 1 na saída *dir* e calcula o inverso do módulo de *omega* para usar como período. Se a velocidade for negativa, o gerador coloca 0 na saída *dir*. Então, o gerador calcula a integral do tempo e compara com o valor do período calculado a partir da velocidade de entrada. Quando a integral é maior que o período calculado, é gerado um pulso na saída *pulso* e a integral do tempo reiniciada. Este modelo é utilizado por ser semelhante ao “estouro” do *timer* de microcontrolador PIC.

O cálculo da saída é feito através da comparação de três valores:

V_{PD} – Velocidade de saída calculado por uma lei de controle PD;

V_{MAX} – Velocidade de saída máxima que não ultrapassa a aceleração máxima;

V_{MIN} – Velocidade de saída mínima que não ultrapassa a desaceleração máxima.

Se V_{PD} for maior que V_{MAX} , então a saída é dada por V_{MAX} . Se V_{PD} for menor que V_{MIN} , então a saída é dada por V_{MIN} . Se V_{PD} estiver entre V_{MAX} e V_{MIN} , então a saída é dada por V_{PD} . A saída deve estar dentro de um limite para evitar que o motor atinja uma velocidade excessiva. Este algoritmo é feito a partir da base de regras proposta no Capítulo 4. A seção seguinte apresenta estes modelos integrados para a simulação do sistema motor-controlador.

7.5. Simulação de um motor com controlador

A simulação do sistema com os motores e o controlador proposto é feita a partir dos modelos apresentados. A Figura 53 apresenta o modelo implementado em Simulink.

Os parâmetros do motor são os apresentados na Figura 47. O modelo do *encoder* é apresentado na Figura 54. O *encoder* deste modelo possui *6000 ppr* (pulsos por revolução).

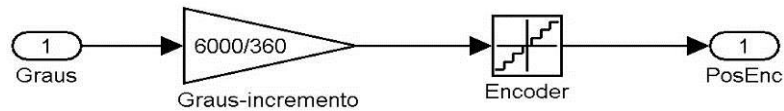


Figura 54 – Modelo do *encoder*

Neste sistema, o motor está ligado a um redutor com relação de transmissão de *20:1*. A posição demandada para o sistema é dada em relação aos incrementos do *encoder*. Primeiro é apresentado o resultado da simulação de um degrau na posição demandada. É demandada na entrada a posição *250*, ou seja, o *encoder* deve realizar *250* pulsos de rotação no sentido positivo, o que resulta em uma rotação de *15°* no *encoder*. Como entre o *encoder* e o motor existe o redutor de *20:1*, o motor deve realizar uma rotação de *300°*. A Figura 55 apresenta o resultado da simulação.

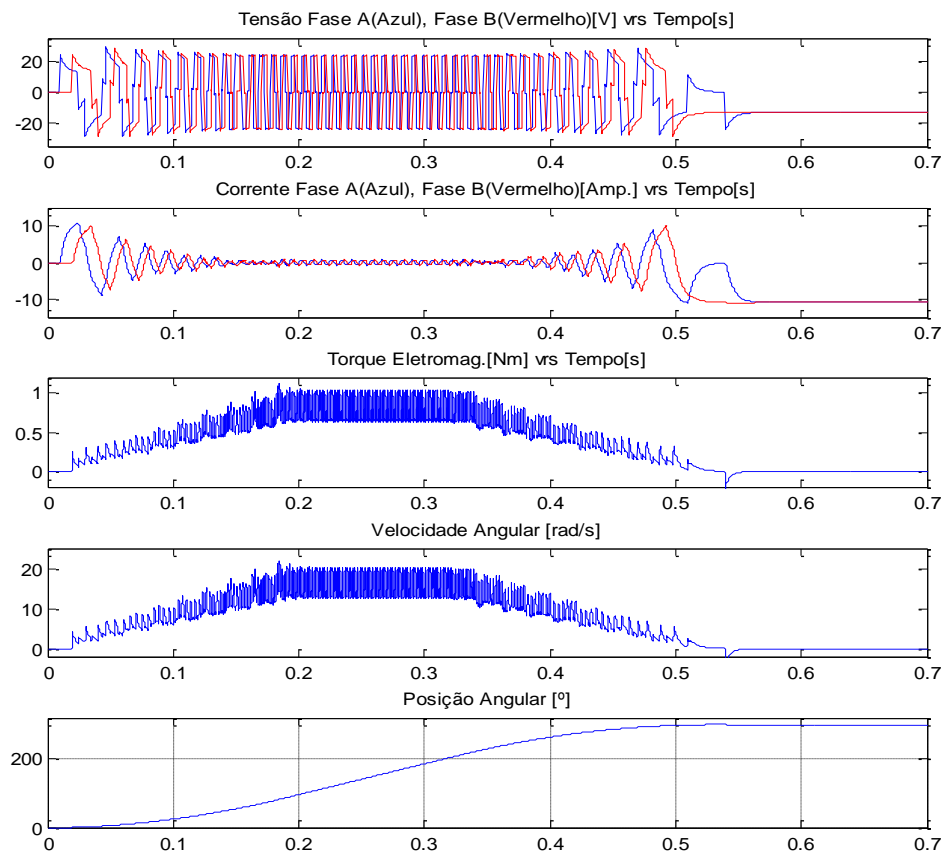


Figura 55 – Gráficos do motor

Os gráficos mostram que o motor acelera entre aproximadamente os instantes 0 e $0,2s$, possui velocidade constante entre $0,20s$ e $0,32s$, e desacelera entre $0,32s$ e $0,5s$. O torque eletromagnético gerado alcançou o pico de cerca de $1Nm$. E a velocidade alcançou $20,2rad/s$ de pico. A frequência máxima do gerador de pulsos é de 1000 passos/segundo, o que resulta em um período de $0,001s$. Como o motor utilizado realiza $400ppr$ em modo de excitação meio-passo, a cada pulso na entrada do *driver* o motor deve girar $0,9^\circ$. Logo, a velocidade máxima do motor deve ser de aprox. $15,7rad/s$. Ao observar o gráfico da Figura 56, pode ser observado que a velocidade quando o controlador demanda a velocidade máxima varia entre $20,2rad/s$ e $12,6rad/s$. Isto ocorre devido à natureza oscilatória do motor de passo.

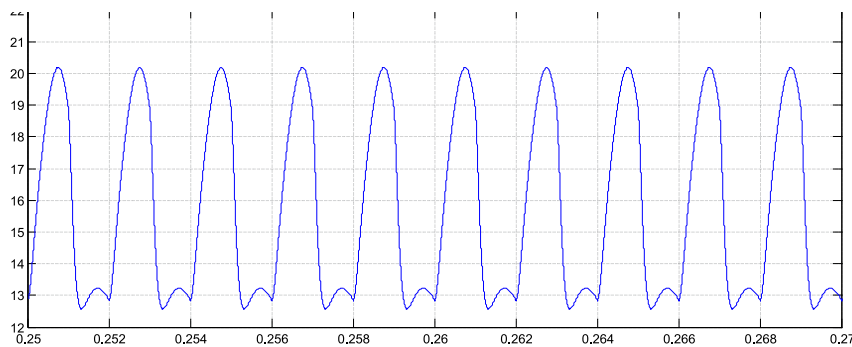


Figura 56 – Velocidade máxima do motor

A Figura 57 mostra a posição do *encoder* ao longo do tempo. Pode ser observado que o motor alcança o valor demandado em aproximadamente $0,5s$.

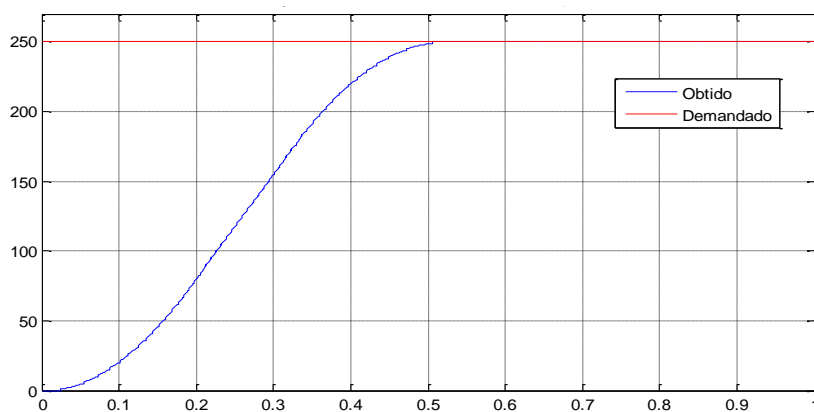


Figura 57 – Posição do *encoder*

Em seguida são apresentados os resultados para o motor sendo demandado de uma trajetória senoidal. A Figura 58 apresenta o resultado para a frequência de $0,05\text{Hz}$ e amplitude de 40 incrementos do *encoder*.

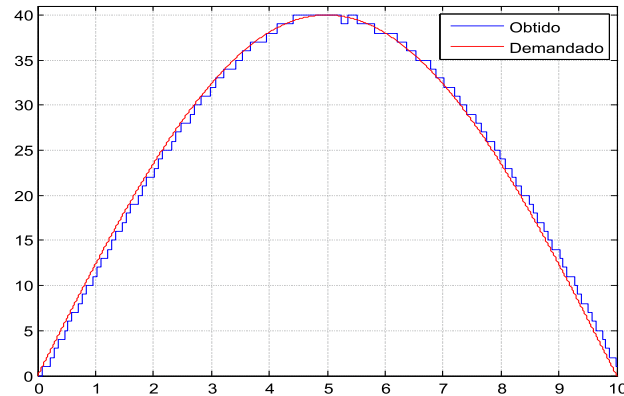


Figura 58 – Sistema demandado de 40inc. de amplitude, a $0,05\text{Hz}$

No caso apresentado na Figura 58, o sistema é capaz de seguir a trajetória com um erro próximo à resolução do *encoder*. Logo, o erro observado é devido em grande parte à discretização na medida do *encoder*. A Figura 59 apresenta o resultado para a mesma amplitude, mas uma frequência dez vezes maior, ou seja, $0,5\text{Hz}$. O erro aumentou devido à maior velocidade demandada.

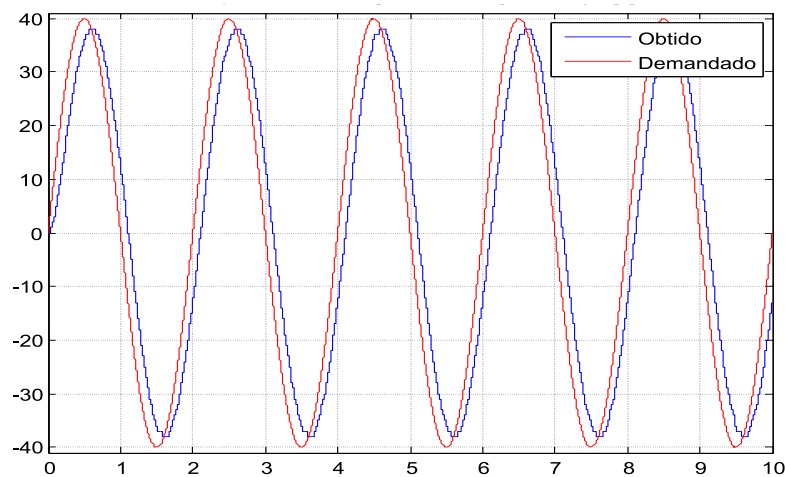


Figura 59 – Sistema demandado de 40inc. de amplitude, a $0,5\text{Hz}$

A seção a seguir mostra como o controlador deve ser ajustado.

7.6. Ajuste do controlador

Na seção 4.3 são propostos alguns ensaios para determinar as constantes α e V_s . A partir destas constantes, a eq. (4.12) fornece uma estimativa para o ganho

proporcional K_p . O ganho K_D pode ser estimado inicialmente com um valor quase nulo. O ajuste fino das constantes K_p e K_D pode ser feito observando o comportamento do sistema quando a saída do controlador usa o valor calculado pela lei PD, V_{pd} . A simulação do degrau apresentada na seção anterior possui $\alpha = 5000$ e $V_s = 1000$. A partir destes valores, a eq. (4.12) estima $K_p \cong 10$. A constante proporcional pode ter um valor baixo, então é estimado por $K_D = 0,02$. A Figura 60 mostra, para este caso, o controlador usando o valor calculado pelo PD.

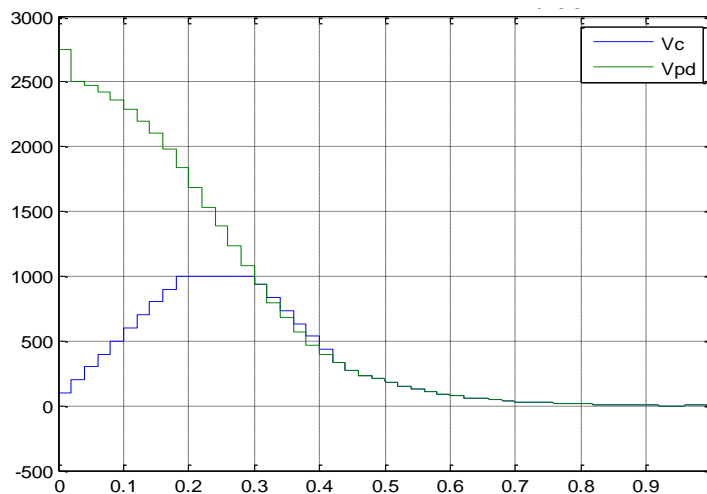


Figura 60 – Saída com $K_p = 10$ e $K_D = 0,02$

A linha V_c é a saída do controlador, enquanto V_{pd} é o valor calculado pela lei de controle PD. Nesta condição, a desaceleração é muito lenta e o sistema demora a alcançar a meta. Se o motor não estiver perdendo passos, o valor de V_c corresponde a aproximadamente a velocidade do motor (em *passos/s*, ou pps). Como o valor de K_p é baixo, o controlador usa o valor calculado pelo PD durante grande parte da desaceleração. Entretanto, é provável que o sistema consiga desacelerar da mesma forma como acelerou, então o valor de K_p pode ser aumentado. A Figura 61 apresenta o resultado da saída do controlador se $K_p = 15$.

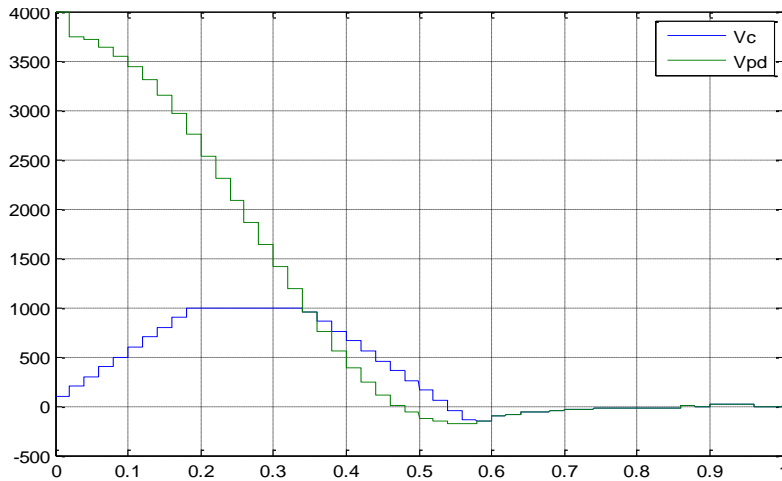


Figura 61 – Saída com $K_p = 15$ e $K_D = 0,02$

Nesta condição ocorre uma resposta exagerada do controlador e o sistema ultrapassa a meta, tendo que inverter o movimento para alcançar o valor correto, ou seja, um *overshoot*. A desaceleração neste caso é semelhante à aceleração, mas o sistema começa a desacelerar muito tarde. Devido à base de regras, o instante em $t \cong 0,35s$, onde $V_{PD} = V_C$, determinou o início da desaceleração. A Figura 62 apresenta o resultado da saída do controlador se $K_p = 13$.

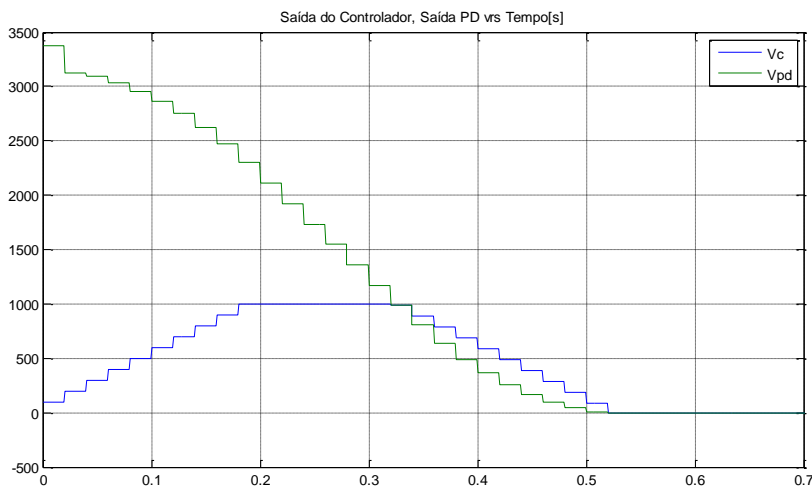


Figura 62 – Saída com $K_p = 13$ e $K_D = 0,02$

Neste caso, a saída do controlador é igual ao calculado pelo PD em dois momentos: no início da desaceleração e no final do movimento. Entre os instantes $t=0$ e $t=0,18s$ o motor acelera. Esta aceleração é aproximadamente igual a α , por que ao longo deste intervalo $V_{PD} \geq V_{MAX}$. No instante $t=0,18s$ a saída é igual à máxima, ou seja, ocorre a saturação da saída do controlador. E entre os instantes $t=0,18s$ e $t=0,32s$ o motor mantém uma velocidade constante próxima a

V_S . Em $t=0,32s$ o motor começa a desaceleração. Isto ocorre porque $V_{MIN} < V_{PD} < V_S$. Entre os instantes $t=32s$ e $t=52s$ o sistema desacelera. O valor da desaceleração é aproximadamente α , por que $V_{PD} \leq V_{MIN}$. A partir de $t=0,52s$ o erro é nulo e a saída do controlador também. Este ajuste pode ser considerado bom, pois o motor realizou um perfil de velocidade trapezoidal, onde a aceleração é próxima da desaceleração. Se a aceleração estiver próxima da máxima, então a desaceleração também estará. Logo, o motor opera próximo das condições ótimas.

Nestes ajustes é suposto que a velocidade do motor é aproximadamente a demandada pelo controlador. A Figura 63 apresenta, em um mesmo gráfico, a velocidade do motor e a demandada pelo controlador.

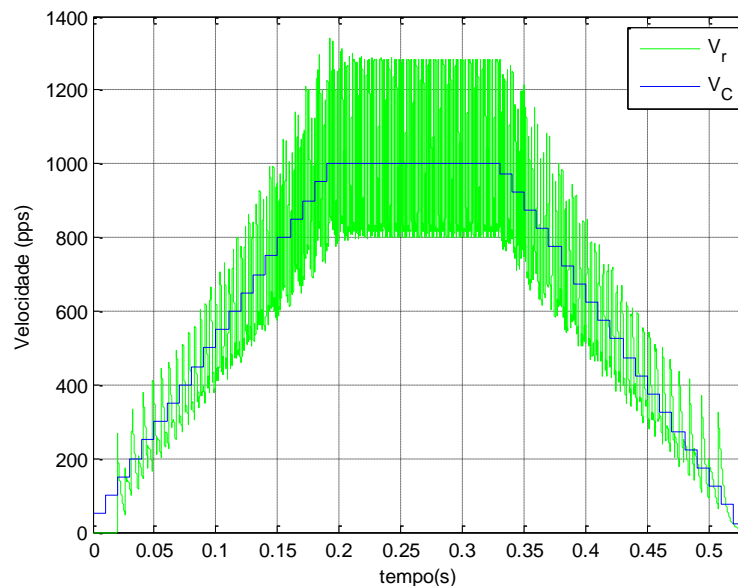


Figura 63 – Velocidade do motor vs. demandada

Devido à natureza oscilatória do motor de passo, a velocidade angular oscila muito. Entretanto, é possível notar que a saída do controlador V_C corresponde a aproximadamente a velocidade média do motor.

Na seção seguinte é apresentada a simulação dos motores de passo no manipulador proposto.

7.7. Simulação do manipulador

O modelo do manipulador utiliza a equação de movimento, eq. (6.52), para calcular os torques dos atuadores. A Figura 64 apresenta o modelo implementado no Simulink.

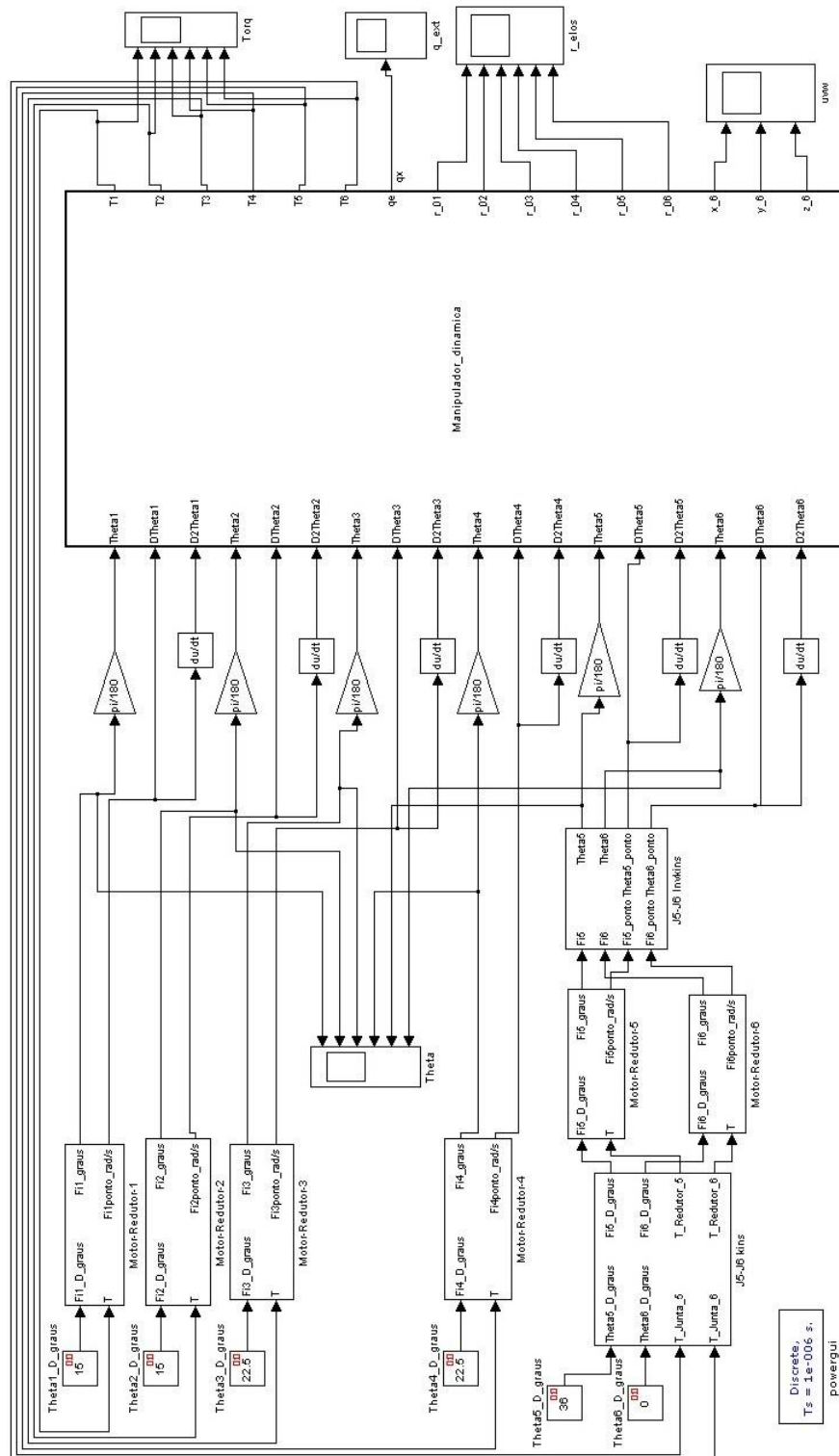


Figura 64 – Modelo com motores e manipulador

A função “*manipulador_dinamica*” recebe os ângulos, velocidades e acelerações das juntas e calcula o torque necessário em cada junta para realizar este movimento. Nos modelos dos motores de cada junta, a entrada de torque da

carga recebe este torque calculado. Esta função também calcula a posição e orientação da extremidade. Estes dados são calculados para geração de gráficos. O código em MATLAB desta função é apresentado no Apêndice B. A função “*M_gen*” calcula a matriz de inércia do manipulador a partir das coordenadas das juntas.

Os blocos “*Motor-Redutor*” incluem os modelos dos motores e controladores de cada junta, como apresentado anteriormente. As diferenças entre os blocos de cada junta são apresentados abaixo.

- Parâmetros do motor;
- Modo de excitação do motor;
- Relação de transmissão dos redutores;
- Resolução dos *encoders*;
- Parâmetros do controlador.

Os parâmetros dos motores estão no Apêndice C. A Tabela 5 possui os demais dados de cada bloco.

Tabela 5 – Configuração dos blocos *Motor-Redutor*

Bloco	Modo Excitação	Transmissão	<i>Encoder</i>	K_p	K_D
1	Meio Passo	20:1	6000ppr	13	0,02
2	Meio Passo	20:1	6000ppr	13	0,02
3	Meio Passo	20:1	4000ppr	19	0,02
4	Meio Passo	10:1	4000ppr	10	0,02
5	Passo Completo	10:1	2500ppr	15	0,02
6	Passo Completo	10:1	2500ppr	15	0,02

O coeficiente de atrito do motor utilizado na simulação é estimado em função dos mancais, mas não é um valor crítico para a simulação. E em muitos casos o atrito do motor é desprezível comparado com o da carga. Em alguns dos motores simulados, o valor torque de retenção não é fornecido pelo fabricante. Então é estimado como 5% do torque de fixação, que é um parâmetro fornecido. O fluxo máximo induzido é um parâmetro não fornecido pelo fabricante, mas é importante para a simulação. Assim, é necessário calcular este valor a partir de um ensaio do motor. A seção seguinte mostra como este cálculo é feito a partir de uma instrumentação do motor.

7.8. Cálculo do fluxo máximo induzido dos motores utilizados

O fluxo máximo induzido Ψ_M nos motores não é fornecido pelos fabricantes dos motores testados. O cálculo de Ψ_M é feito através da eq. (3.16). Logo, é necessário medir a tensão em uma das fases com o circuito aberto e a velocidade angular do rotor. A Figura 65 apresenta o esquema da bancada utilizada para medir tensão e velocidade angular dos motores. A bancada consiste de um torno onde o eixo do motor de passo é fixado à placa universal. Desta forma, é gerada uma velocidade constante no rotor do motor de passo. Um tacômetro é acoplado ao eixo do rotor. O osciloscópio mede a forma de onda gerada em um dos enrolamentos do motor com o circuito aberto.

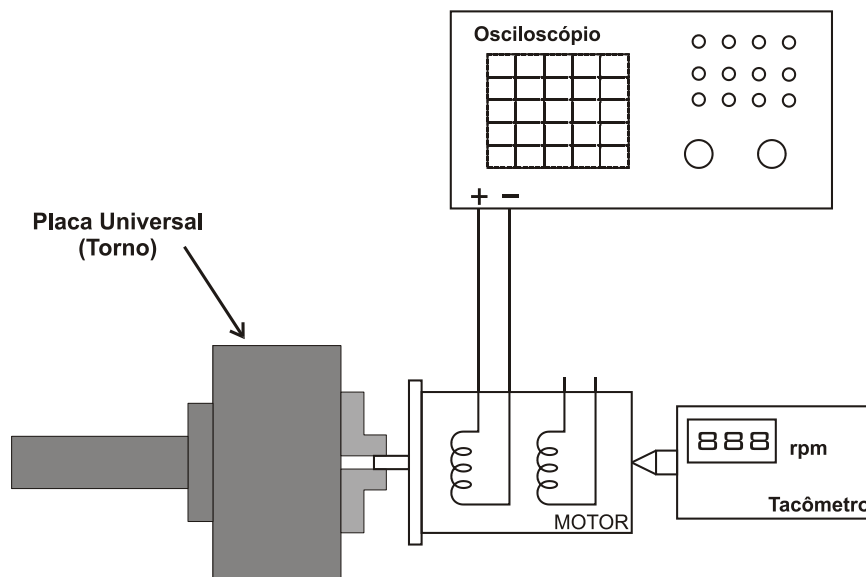


Figura 65 – Esquema da bancada de medidas

A Figura 66 apresenta uma foto da bancada de medidas, onde é possível ver o motor girando, o tacômetro medindo a velocidade de rotação e o osciloscópio mostrando a forma de onda. Para três velocidades de rotação, a máxima tensão alternada medida no osciloscópio é usada no cálculo do fluxo induzido máximo.



Figura 66 – Foto da bancada de medidas

A Figura 67 apresenta uma foto do osciloscópio e um esquema de como a tensão máxima é medida.

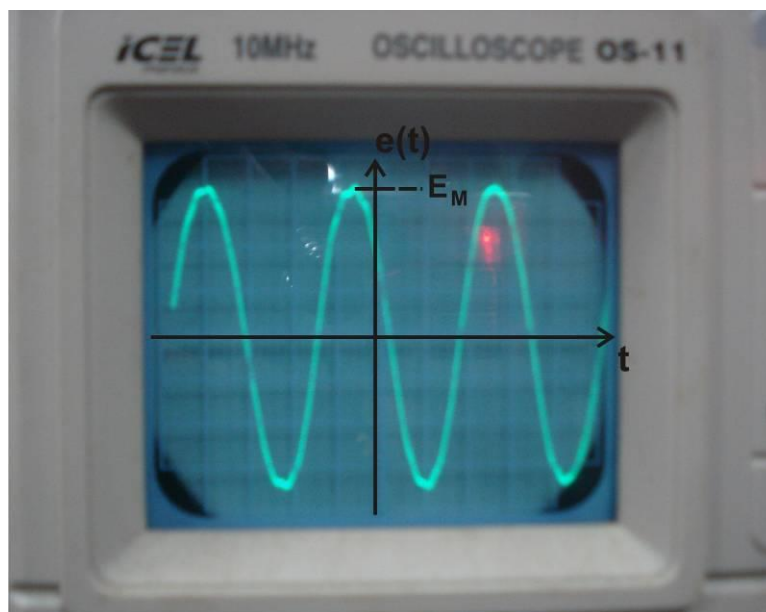


Figura 67 – Medida de tensão no osciloscópio

O torno mecânico é acionado em três velocidades distintas, gerando assim um valor de tensão máxima para cada condição. A Tabela 6 apresenta os valores medidos e o fluxo induzido máximo calculado através da eq. (3.16).

Tabela 6 – Cálculo do Fluxo Máximo Induzido

Motor	N(RPM)	E_M (V)	θ_s (°)	m	p	Ψ_M (Vs)	Ψ_M (Vs) média
AK85H/3.75-1.8	70,1	7,2	1,8	2	50	0,0196	0,019
AK85H/3.75-1.8	117,5	11,8	1,8	2	50	0,0192	
AK85H/3.75-1.8	204,4	20	1,8	2	50	0,0187	
AK85H8/3.36-1.8	70,1	5,4	1,8	2	50	0,0147	0,014
AK85H8/3.36-1.8	117,5	9	1,8	2	50	0,0146	
AK85H8/3.36-1.8	204,4	15	1,8	2	50	0,0140	
AK57H/3Gxx-1.8	70,1	3,5	1,8	2	50	0,0095	0,010
AK57H/3Gxx-1.8	117,5	6,8	1,8	2	50	0,0111	
AK57H/3Gxx-1.8	204,4	10	1,8	2	50	0,0093	
AK56H/3-1.8	70,1	3	1,8	2	50	0,0082	0,0094
AK56H/3-1.8	117,5	5,6	1,8	2	50	0,0091	
AK56H/3-1.8	204,4	11,6	1,8	2	50	0,0108	

Desse modo, todos os parâmetros do motor de passo são conhecidos.

7.9.

Resultado da simulação do manipulador

Na simulação do manipulador com motores de passo e o controle proposto, é demandado o vetor posição das juntas $\bar{q} = [15^\circ; 15^\circ; 22,5^\circ; 22,5^\circ; 36^\circ; 0^\circ]$. O manipulador parte da posição onde todos os ângulos das juntas são nulos. A partir da cinemática direta, pode-se determinar que a extremidade do manipulador parte da posição $(0,350; 0,00; -0,210)m$ e deve alcançar a posição aproximada $(0,4714; 0,1147; -0,0778)m$. A Figura 68 mostra o gráfico do erro da posição da extremidade ao longo do tempo.

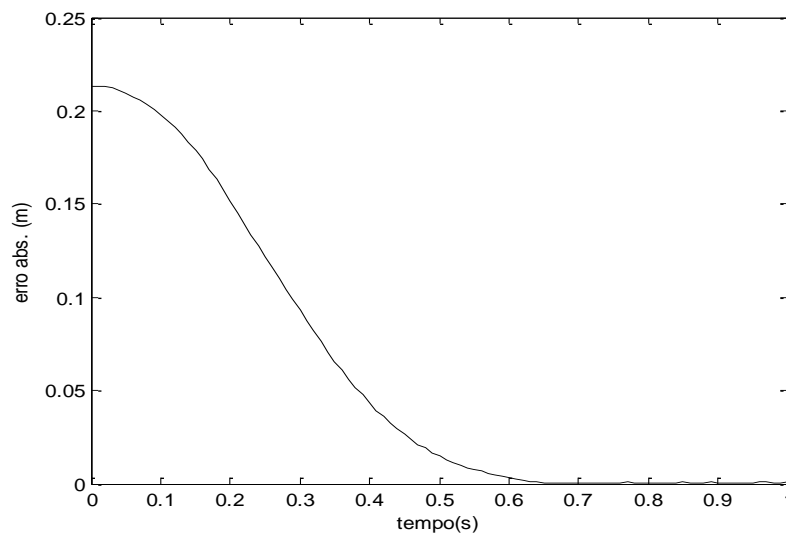


Figura 68 – Erro da posição da extremidade

O gráfico do erro ao longo do tempo mostra que o manipulador demora cerca de $0,65s$ para alcançar a posição desejada. Ao final do movimento, o erro de

posicionamento da extremidade é de cerca de $0,5\text{mm}$. A Figura 69 mostra a posição dos elos do manipulador em vários instantes de tempo. As linhas tracejadas representam os elos e o círculo a extremidade.

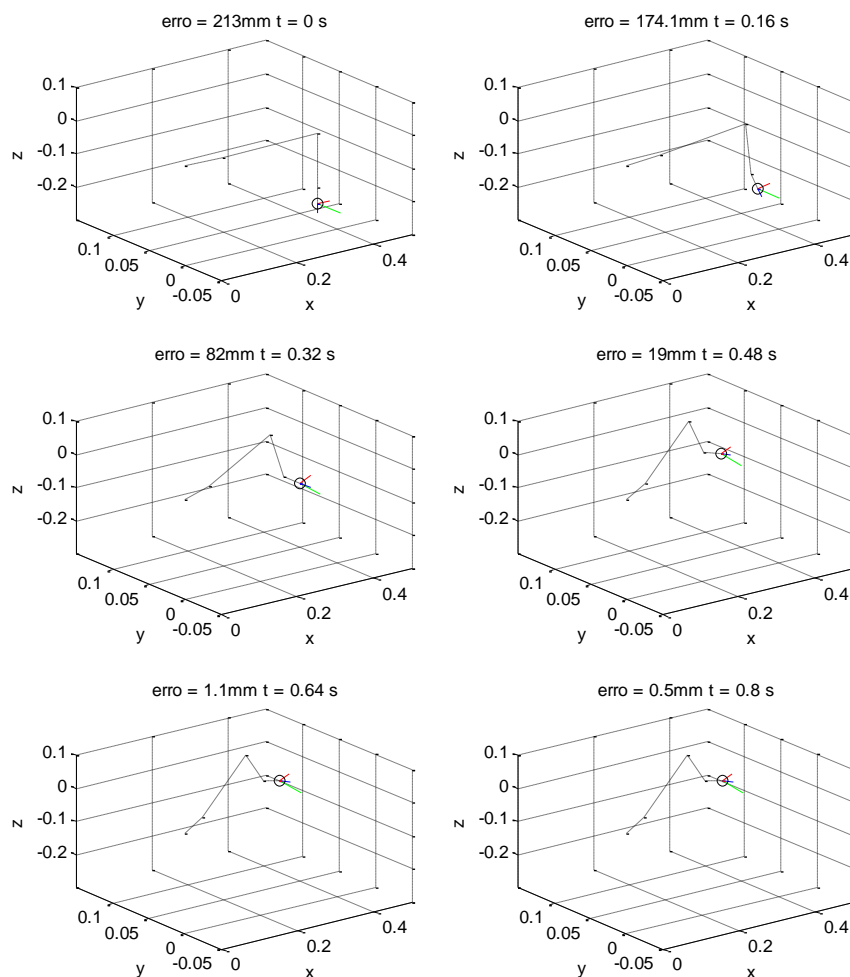


Figura 69 – Movimento do manipulador

A Figura 70 mostra o rastro do movimento da extremidade.

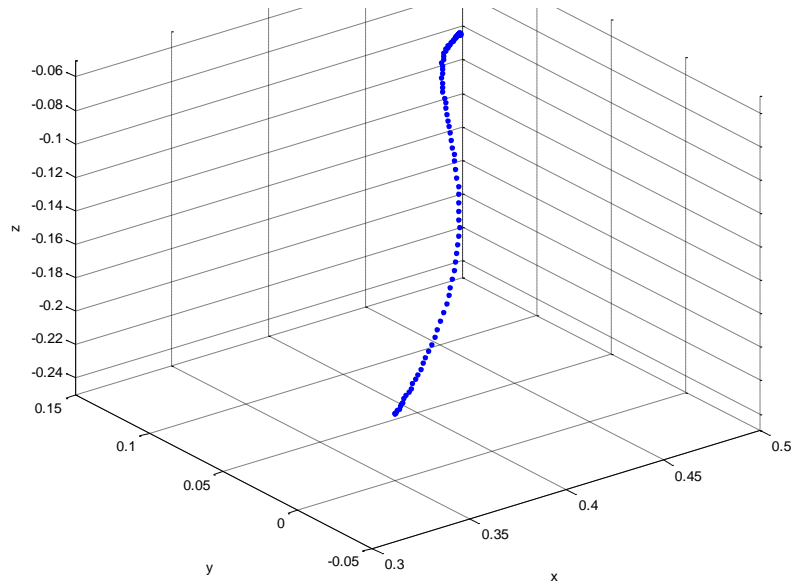


Figura 70 – Rastro do manipulador

As simulações mostraram que o desempenho do manipulador é adequado para suas dimensões. O erro do posicionamento final é devido, principalmente, ao erro de quantização do *encoder* e a resolução do motor de passo.

8 Experimentos

O manipulador usando motores de passo e o controle proposto é testado para medir a precisão de seu posicionamento. A precisão absoluta avalia a capacidade do manipulador de se aproximar de uma posição no seu espaço de trabalho. A repetibilidade avalia a precisão do retorno do manipulador a um ponto qualquer [30]. Medidas de precisão absoluta e repetibilidade são feitas. A partir destes resultados, o controle proposto é validado.

8.1. Calibração do sistema

Primeiro, a base do manipulador é nivelada com um nível de precisão. Este procedimento é feito para obter uma referência precisa da posição dos elos e da orientação da extremidade. A Figura 71 apresenta o manipulador com o nível de precisão.

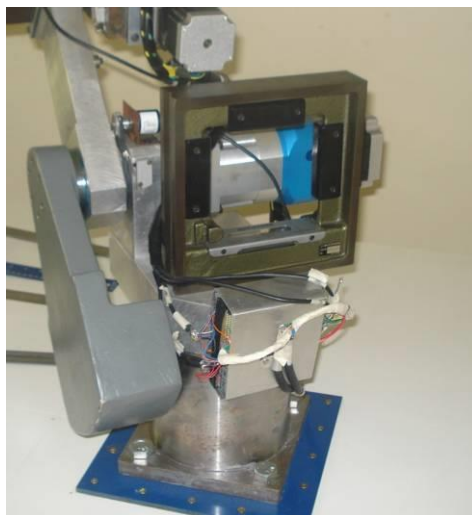


Figura 71 – Nivelamento da base

Após o nivelamento, o erro máximo obtido é de $0,02\text{mm/m}$ na base. Em seguida, um nível comum é posto no elo 2 para obter a posição angular da junta 2 onde $\theta_2 = 0$. A Figura 72 mostra este procedimento.

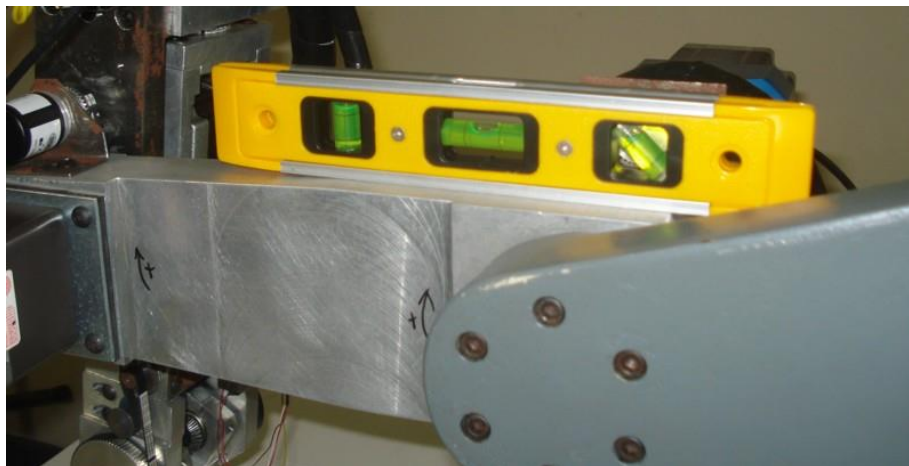


Figura 72 – Calibração da junta 2

Após observar que a junta 2 está nivelada. O botão *reset* da junta 2 é acionado, assim o controlador é informado de que está na posição inicial.

As juntas 3 e 4 são calibradas juntas, com o manipulador posicionado em $\theta_2 = 90^\circ$. O nível é observado em duas direções, uma relativa à junta 3 e outra à junta 4. A Figura 73 mostra a calibração dessas juntas.

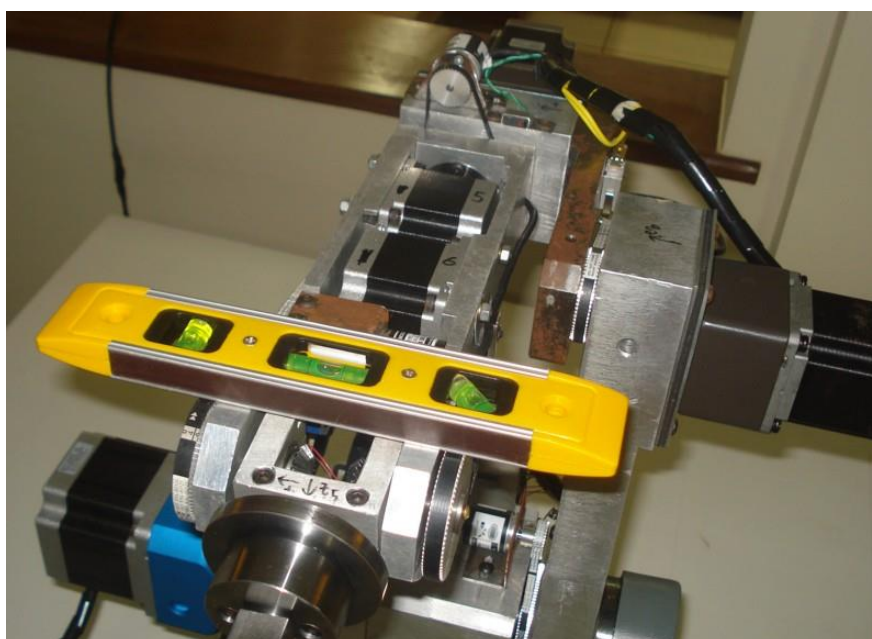


Figura 73 – Calibração da juntas 3 e 4

Nesta mesma posição, um dispositivo é fixado na flange da extremidade para auxiliar na calibração das juntas 5 e 6. Este dispositivo consiste de uma barra quadrada presa a um suporte. Análogo ao procedimento das juntas 3 e 4, o nível é

verificado em duas direções. A Figura 74 mostra este dispositivo sendo usado para calibrar as juntas 5 e 6.

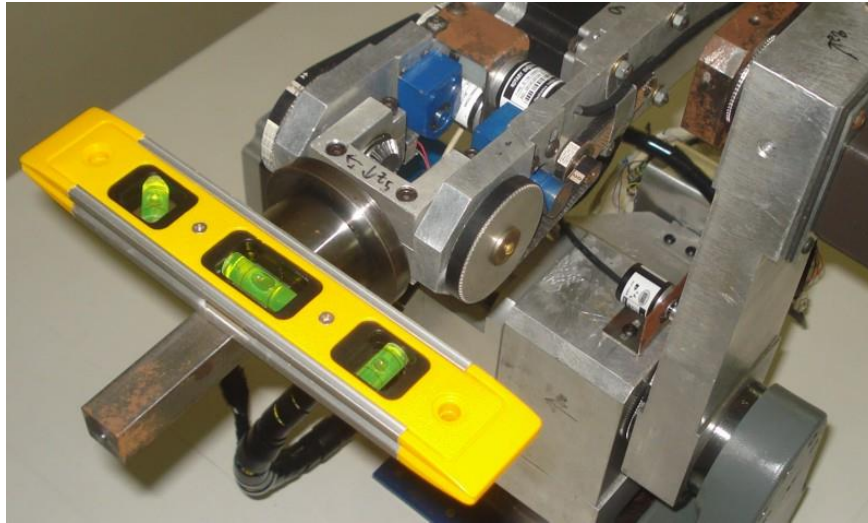


Figura 74 – Calibração das juntas 5 e 6

A junta 1 é calibrada posicionando a extremidade, com as juntas 4 e 6 na posição nula, sobre uma marca de referência na mesa do manipulador. Esta marca é feita com um esquadro de centro posicionado para passar pelo centro da base. A Figura 75 mostra a junta 1 sendo calibrada.

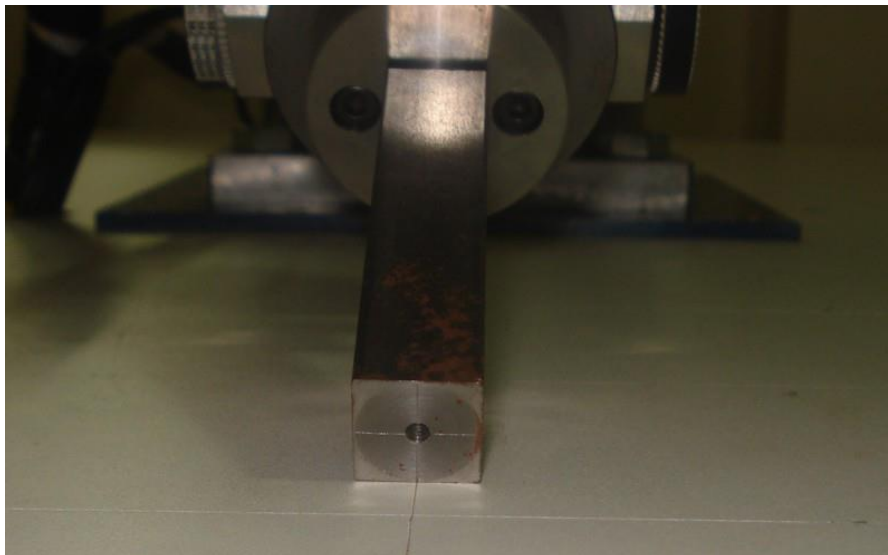


Figura 75 – Calibração da junta 1

A partir destes procedimentos, o manipulador é calibrado na posição de referência, onde todos os ângulos das juntas são nulos. Como a base está nivelada, um nível pode ser usado para medir a orientação da extremidade.

Devido à folga dos redutores, o manipulador oscila em torno da posição demandada com o controle proposto. O controlador é reprogramado para admitir um erro unitário na posição do *encoder* (procedimento conhecido como *anti-chattering*), assim este problema é atenuado. As oscilações deixam de ocorrer quando o sistema está muito próximo da posição desejada, e os ganhos do controlador são reduzidos. Após estes ajustes, o manipulador pode ser testado sem problemas. O programa apresentado no Apêndice A contém estas alterações.

8.2. Sistema de medição

A linha que passa pelo centro do manipulador, utilizada na calibração da junta 1, é utilizada como eixo x de medidas. Uma outra linha, perpendicular a esta, onde $x=200mm$, também é marcada. A Figura 76 mostra estas linhas. Apenas estas duas linhas são utilizadas para as medições. As demais linhas marcadas na mesa são apenas auxiliares, não sendo utilizadas nas medidas. Após uma posição ser marcada na mesa, com um paquímetro, são medidas as distâncias até as linhas de referência. Assim têm-se as posições em x e y de um ponto.

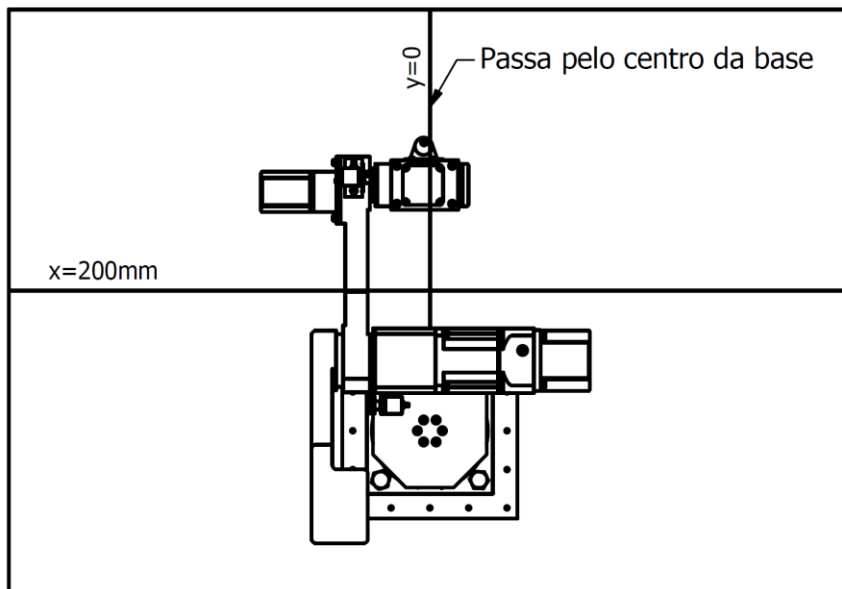


Figura 76 – Linhas de referência

A medida da distância em z é feita com um paquímetro. O paquímetro mede a distância da extremidade até a mesa, então é descontada a altura da mesa até o plano onde $z=0$ ($301,35\text{ mm}$). A Figura 77 mostra como a altura de um ponto é medido em relação à mesa.

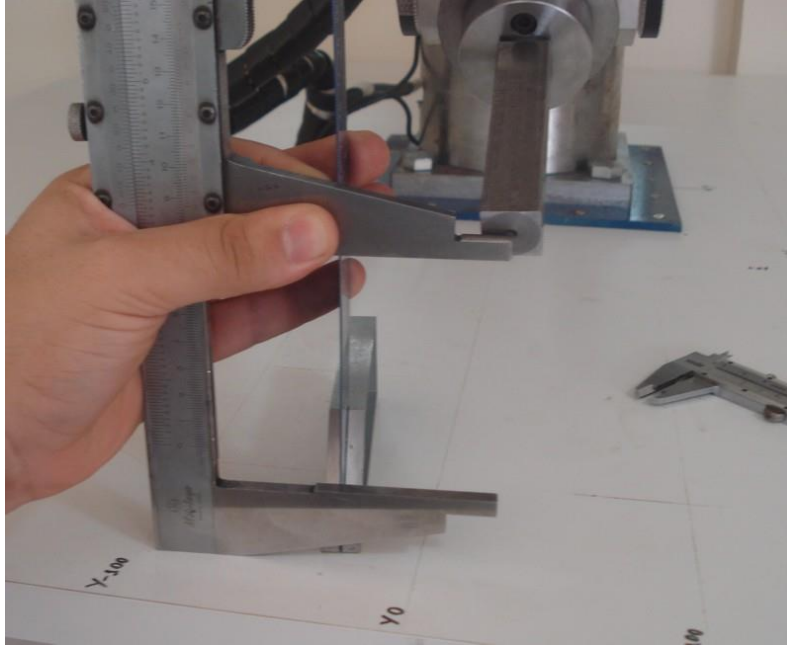


Figura 77 – Media da altura de um ponto

A medida da orientação da extremidade é feita com um transferidor de ângulos acoplado a um nível. Posicionando a base do transferidor no dispositivo acoplado à extremidade, é possível medir o ângulo em relação a horizontal de dois eixos. A Figura 78 mostra o transferidor posicionado na extremidade.

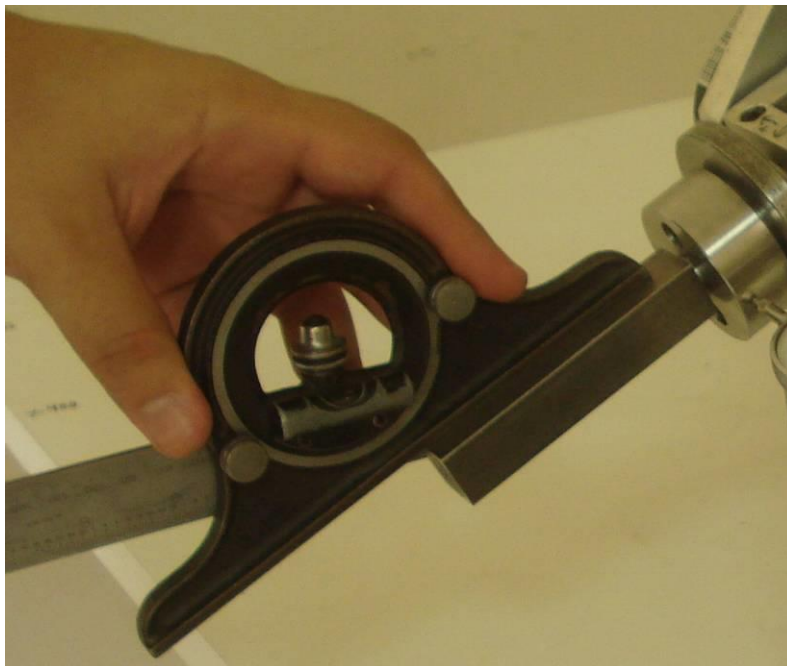


Figura 78 – Medida de orientação da extremidade

Assim, podem ser medidos dois ângulos que a extremidade faz com o plano horizontal. O ângulo feito com o plano vertical pode ser medido projetando o dispositivo na mesa, através de esquadros.

O sistema de medida apresentado mede a posição e orientação da extremidade. Um sistema adicional é utilizado para medir a repetibilidade, mas em apenas uma direção. Apesar de unidirecional, é mais preciso que o sistema proposto anteriormente. A Figura 79 apresenta este dispositivo.



Figura 79 – Medida com relógio comparador

Este sistema consiste de um relógio comparador em uma base magnética. Como a mesa é de madeira, esta base magnética é fixada a uma anilha de ferro fundido. O relógio tem resolução nominal de $0,01mm$, mas medidas da ordem de milésimos de mm podem ser estimadas.

8.3. Posições medidas

Três posições são utilizadas para realizar as medidas. Para simplificar as medidas, a orientação desejada da extremidade é sempre a mesma. Esta orientação possui os vetores unitários:

$$u = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (8.1)$$

$$v = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \quad (8.2)$$

$$w = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (8.3)$$

A primeira posição de medidas possui coordenadas $x=250mm$, $y=-320mm$ e $z=200mm$. Nesta posição, pela cinemática inversa, a posição angular de cada junta é

$$q = [-57,995^\circ; 61,674^\circ; 21,122^\circ; -85,518^\circ; 58,277^\circ; 81,522^\circ] \quad (8.4)$$

Este cálculo é feito com a rotina em MATLAB para cálculo da cinemática inversa apresentada no Apêndice B.

A segunda posição de medidas possui coordenadas $x=400mm$, $y=0mm$ e $z=-160mm$. Neste caso, a posição angular de cada junta é

$$q = [0^\circ; 0^\circ; 0^\circ; 0^\circ; 90^\circ; 0^\circ] \quad (8.5)$$

As coordenadas $x=350mm$, $y=300mm$ e $z=-50mm$ correspondem à terceira posição de medidas. A partir da solução da cinemática inversa, a posição angular de cada junta é

$$q = [45,000^\circ; 19,456^\circ; 14,143^\circ; 50,208^\circ; 66,965^\circ; -25,163^\circ] \quad (8.6)$$

8.4. Cálculo do erro

O erro absoluto de posicionamento da extremidade do manipulador é calculado por

$$erro(abs) = \sqrt{(x_m - x_d)^2 + (y_m - y_d)^2 + (z_m - z_d)^2} \quad (8.7)$$

onde o índice m representa o valor medido e d o valor demandado.

O erro de repetibilidade é calculado por

$$erro(rep) = \sqrt{(x_m - x_{ref})^2 + (y_m - y_{ref})^2 + (z_m - z_{ref})^2} \quad (8.8)$$

onde o índice ref representa o valor de referência, no caso, o primeiro valor medido.

8.5. Resultados

A Tabela 7 apresenta os resultados das medidas na primeira posição.

Tabela 7 – Medidas na primeira posição

x(mm)	y(mm)	z(mm)	Erro (abs)	Repet.
250,5	-320,3	200,4	0,7	
251,1	-320,5	200,9	1,5	0,8
250,5	-320,2	200,5	0,7	0,1
250,7	-320,3	201,1	1,3	0,7
250,1	-319,7	200,2	0,4	0,7
250,3	-320,4	200,5	0,7	0,2
250,9	-320,3	200,3	1,0	0,4
251,0	-319,9	200,3	1,0	0,6
250,6	-319,8	200,4	0,7	0,5
250,2	-320,3	200,2	0,4	0,4

O erro de posicionamento absoluto máximo é de $1,3\text{mm}$. O erro máximo na repetibilidade é de $0,8\text{mm}$. A Tabela 8 apresenta as medidas da repetibilidade com relógio comparador, portanto unidirecional.

Tabela 8 – Medida na primeira posição com relógio comparador

Relógio(mm)
0,281
-0,243
-0,259
0,131
-0,155
0,253
0,127
-0,156
-0,250
-0,111

O relógio é colocado em uma posição aleatória. As medidas estão entre $0,281\text{mm}$ e $-0,259\text{mm}$. A Tabela 9 apresenta os resultados das medidas feitas na segunda posição de medidas.

Tabela 9 – Medidas na segunda posição

x(mm)	y(mm)	z(mm)	Erro (abs)	Repet.
400,4	0,1	-160,2	0,5	
400,8	-0,5	-159,8	1,0	0,8
400,5	0,1	-160,5	0,7	0,3
399,9	0,3	-159,9	0,3	0,6
399,5	-0,3	-160,3	0,7	1,0
400,2	0,0	-160,2	0,3	0,2
400,7	0,4	-160,4	0,9	0,5
400,8	0,2	-160,5	1,0	0,5
400,6	0,1	-160,3	0,7	0,2
400,3	0,5	-160,4	0,7	0,5

Nesta posição, o erro máximo de posicionamento absoluto é de $1,0mm$. O erro máximo na repetibilidade é de $0,8mm$. A Tabela 10 apresenta a medida da repetibilidade com relógio comparador.

Tabela 10 – Medida na segunda posição com relógio comparador

Relógio(mm)
0,050
0,144
0,100
-0,291
-0,251
-0,020
0,071
0,230
-0,036
-0,263

As medidas estão entre $0,230mm$ e $-0,291mm$. A Tabela 11 apresenta os resultados das medidas na terceira posição de medidas.

Tabela 11 – Medidas na terceira posição

x(mm)	y(mm)	z(mm)	Erro (abs)	Repet.
350,4	300,2	-49,8	0,5	
350,9	299,7	-50,2	1,0	0,8
350,7	300,2	-49,8	0,8	0,3
350,5	300,1	-49,7	0,6	0,2
350,3	299,9	-50,2	0,4	0,5
350,4	300,5	-49,5	0,8	0,4
350,2	300,3	-49,8	0,4	0,2
350,3	300,2	-50,1	0,4	0,3
350,4	300,1	-50,3	0,5	0,5
350,3	300,3	-50,4	0,6	0,6

O erro máximo de posicionamento absoluto, nesta posição, é de $1,0mm$. O erro máximo na repetibilidade é de $0,8mm$. A Tabela 12 apresenta a medida da repetibilidade com relógio unidirecional.

Tabela 12 – Medida na terceira posição com relógio comparador

Relógio(mm)
-0,030
-0,168
0,243
-0,205
0,030
-0,221
-0,090
0,023
0,298
0,213

O erro máximo e mínimo médios destas três posições consideradas são $0,298mm$ e $-0,221mm$, respectivamente. Tendo como base as medidas em todas as posições, este manipulador possui precisão absoluta de $\pm 1,3mm$. A repetibilidade medida através das referências da mesa indica um erro de $\pm 0,8mm$. Entretanto, medidas com um relógio comparador indicaram um erro máximo de $\pm 0,3mm$ em uma direção. Se o erro em cada direção for igual a este, o erro máximo total na repetibilidade seria de $\pm 0,5mm$. Devido à maior precisão do relógio comparador, esta medida é mais confiável.

O erro de orientação da extremidade é medido com o transferidor acoplado a um nível. Durante os testes, todas as medidas feitas mostraram um erro inferior à menor resolução do instrumento. Logo, não é possível medir com exatidão este

erro. Mas, como este instrumento tem precisão de cerca de $\pm 0,5^\circ$, pode ser afirmado que o erro é inferior a este valor. Considerando o erro de orientação nulo, na medição onde o erro absoluto é máximo, é calculada a cinemática inversa. Subtraindo este resultado da posição demandada, o erro de posicionamento de cada junta é

$$\text{erro}(\text{angular}) = [-0,07^\circ; 0,12^\circ; -0,61^\circ; 0,29^\circ; 0,10^\circ; -0,57^\circ] \quad (8.9)$$

A rotina implementada no controlador admite um erro unitário na posição do *encoder* para maior estabilidade. Somando este ao erro da discretização, o erro de dois incrementos dos *encoders* em cada junta é dado por

$$\text{erro}(\text{encoder}) = [\pm 0,12^\circ; \pm 0,12^\circ; \pm 0,18^\circ; \pm 0,18^\circ; \pm 0,29^\circ; \pm 0,29^\circ] \quad (8.10)$$

Logo, o erro do sistema se deve principalmente ao erro da discretização dos sensores somado ao erro admitido para melhorar a estabilidade.

Estes resultados mostram que o controle de motores de passo proposto apresenta desempenho adequado no equipamento testado.

9 Conclusões e trabalhos futuros

Neste trabalho, foi desenvolvido um algoritmo de controle para motores de passo. Simulações feitas no ambiente Simulink/MATLAB do controle proposto demonstram a efetividade do sistema. Um manipulador robótico de seis graus de liberdade foi proposto para validar a técnica apresentada. O manipulador consiste de seis motores de passo acoplados a redutores de velocidade que movimentam seis juntas rotativas. O alcance máximo do manipulador é de $540mm$. O projeto do manipulador foi feito em *software* CAD. A partir deste *software* foram calculadas as massas dos elos e seus momentos de inércia. Estes dados foram utilizados para escrever a equação do movimento do manipulador. A partir desta equação, os torques nos motores foram calculados em função da velocidade e aceleração angular de cada junta. Um modelo do manipulador com os motores de passo foi feito a partir desta equação de movimento. Este modelo mostrou que o manipulador teria desempenho adequado para suas dimensões.

O manipulador foi então construído e montado seguindo o projeto em CAD. Caixas elétricas contendo os *drivers* dos motores de passo foram montadas. Um computador foi adquirido para enviar dados da trajetória do manipulador para as juntas. Este computador envia os dados através de sua porta paralela e utiliza o *software* EMC2. O EMC2 foi configurado para calcular a cinemática direta e inversa do manipulador, deduzidas nesse trabalho. O *software* possui uma interface gráfica onde é possível programar a trajetória. A posição de cada junta do manipulador é medida com um *encoder* e enviada para um controlador que recebe os dados da posição demandada pelo computador. O controlador envia uma sequência de pulsos para o *driver* do motor de passo. Este controlador foi construído com base do microcontrolador PIC18F2431.

Alguns testes foram feitos para avaliar a precisão absoluta e a repetibilidade do manipulador. Estes testes mostram que o manipulador possui uma precisão absoluta de cerca de $\pm 1,3mm$. A repetibilidade do posicionamento da extremidade foi de cerca de $\pm 0,5mm$. Na simulação, o manipulador alcançou a posição

demandada com erro de $0,5mm$. Este erro, como o medido no teste de repetibilidade, foi devido principalmente à discretização do *encoder*. O erro na precisão absoluta foi maior devido a erros no processo de fabricação.

A precisão do posicionamento angular foi inferior a $\pm 0,5^\circ$. Durante os testes, foi observado que devido à folga dos redutores, o manipulador oscilava em torno da posição demandada. Ajustes “*anti-chattering*” foram feitos nos controladores de modo que admitissem um erro unitário na medida dos *encoders*. Além disso, quando o sistema está muito próximo da posição desejada, os ganhos do controlador são reduzidos. Após estes ajustes, o manipulador apresentou bom desempenho.

A aceleração e velocidade máxima são constantes determinadas no ajuste do controlador. Entretanto, o desempenho do sistema seria melhorado se o sistema fosse capaz calcular os melhores valores em função da condição atual de utilização. Esta é uma recomendação para trabalhos futuros.

10

Referências bibliográficas

- [1] Ali, H. I.; **Robust QFT Controller Design for Positioning a Permanent Magnet Stepper Motors**; International Journal of Computer and Electrical Engineering, Vol. 1, No. 1; April, 2009; 1793-8198.
- [2] Behal, A.; Feemster, M.; Dawson, D.; Mangal, A.; **Sensorless Rotor Velocity Tracking Control of the Permanent Magnet Stepper Motor**; Proc. of the IEEE International Conference on Control Applications; Alaska; 25-27 September, 2000.
- [3] Gong, S.; He, B.; **LabVIEW-Base Automatic Rising and Falling Speed Control of Stepper Motor**; International Conference on Electrical Machines and Systems; p. 1-4; Tokyo; 15-18 November, 2009.
- [4] Liu, J.; Cheng, K. W. E.; To, S.; **Dual Closed Loop Controller of Bus Stepper Motor Based on Back-EMF**; 3rd International Conference on Power Electronics Systems and Applications; 2009.
- [5] Bendjedia, M.; Ait-Amirat, Y.; Walther, B.; Berthon, A.; **Sensorless Control of Hybrid Stepper Motor**; European Conference on Power Electronics and Applications; 2-5 September, 2007.
- [6] Bendjedia, M.; Ait-Amirat, Y.; Walther, B.; Berthon, A.; **DSP Implementation of Rotor Position Detection Method for Hybrid Stepper Motors**; Power Electronics and Motion Control Conference; p. 1-5; 14-16 August, 2006.
- [7] Obermeier, C.; Kellermann, H.; Brandenburg, G.; **Sensorless Field Oriented Speed Control of a Hybrid and a Permanent Magnet Disk Stepper Motor Using an Extended Kalman Filter**; Electric Machines and Drives Conference Record; Milwaukee, WI; May, 1997.

- [8] Defoort, M.; Nollet, F.; Floquet, T.; Perruquetti, W.; **Higher order sliding mode control of a stepper motor**; Proceeding of the 45th IEEE Conference on Decision & Control; p. 4002-4007; 13-15 December, 2006.
- [9] Shah, C; **Sensorless Control of Stepper Motor Using Kalman Filter**; Cleveland, 2004. 128p.; Dissertação de Mestrado – Cleveland State University.
- [10] Acarnley, P.; **Stepping Motors – A guide to theory and practice**; 4.ed.; London: The Institution of Electrical Engineer; 2002; 159p.
- [11] Crnosija, P.; Ajdukovic, S.; Kuzmanovic, B.; **Microcomputer Implementation of Optimal Algorithms for Closed-loop Control of a Hybrid Stepper Motor Drives**; Proceeding of the IEEE International Symposium on Industrial Electronics; p.679-683, Vol. 2; Bled, Slovenia;12-16 July, 1999.
- [12] Elksasy, M.; Gad, H.; **A New Technique for Controlling Hybrid Stepper Motor Through Modified PID Controller**; International Journal of Electrical & Computer Sciences, IJECS-IJENS; Vol. 10, no. 2; 2010.
- [13] Web site Mecatrônica Fácil; **Linguagem de Programação de Robôs**; Disponível em < <http://www.mecatronicaatual.com.br/secoes/leitura/418>>; Acessado em 21 de Maio de 2010.
- [14] Cao, L.; Schwartz, H.; **Oscillation, Instability and Control of Stepper Motor**, Nonlinear Dynamics 18; p.383-404; Kluwer Academic Publishers'; Netherlands; 1999.
- [15] Kenjo, T.; Sugawara, A.; **Stepping Motors and Their Microprocessor Controls**; 2.ed.; Oxford: Oxford University Press; 2003; 279p.
- [16] Nordin, M.; Galic, J.; Gutman P.; **New Models for Backlash and Gear Play**; International Journal of Adaptive Control and Signal Processing, Vol. 11, 49-63; 1997.
- [17] Wu, X.; Meagher, J.; Sommer, A.; **A Differential Planetary Gear Model with Backlash and Teeth Damage**; Proc. of the Society for Experimental Series; Vol. 8, p.203-215; 2011.

- [18] Hovland, G.; Hanssen, S.; Moberg, S.; Brogardh, T.; Gunnarsson, S.; Isaksson, M.; **Nonlinear Identification of Backlash in Robot Transmissions**; Proceeding of the 33rd International Symposium on Robotics; October 7-11, 2002.
- [19] Hughes, A.; Lawrenson, P. J.; **Electromagnetic Damping in Stepping Motors**; Proc. IEE 122, (8) 819-24; 1975.
- [20] Oswald, A.; Herzog, H.G.; **Invertigation of the Usability of 2D- and 3D- FEM for Hybrid Stepper Motor**; IEEE International Electric Machines and Drives Conference; p. 535-542; Miami, FL; May, 2009.
- [21] Hoang, L.; Brunelle, P.; Sybille, G.; **Design and Implementation of a Versatile Stepper Motor Model for Simulink's SimPowerSystems**; IEEE International Symposium on Industrial Electronics; p. 437-442; Cambridge; 2008.
- [22] Kamalasadán, S.; **A New Intelligent Controller for the Precision Tracking of Permanent Magnet Stepper Motor**; Power Engineering Society General Meeting, p1-7; 24-28 June, 2007.
- [23] Ferrah, A.; Bani-Younes, J. A.; Bouzguenda, M.; Tami, A.; **Sensorless Speed and Position Estimation in a Stepper Motor**; International Conference on Electrical Machines and Power Electronics; p297-302; 2007.
- [24] Austin, D.; **Generate Stepper Motor Speed Profiles in Real Time**; Disponível em: <http://fab.cba.mit.edu/classes/MIT/961.09/projects/i0/Stepper_Motor_Speed_Profile.pdf>. Acesso em: 11 jan. 2012.
- [25] Ogata, K.; **Engenharia de Controle Moderno**; 4ed.; Editora Pearson; 2003; p.800.
- [26] Microchip; **PIC18F2331/2431/4331/4431 Data Sheet**; Disponível em: <www.microchip.com>; 2007.
- [27] Linux.org; **EMC2 User Manual V2.4**; Disponível em <www.linuxcnc.org>; 2010.
- [28] Craig, J. J.; **Introduction to Robotics Mechanics and Control**; 3.ed.; London: Pearson Education; 2005; 400p.

- [29] TSAI, L. W.; **Robot Analysis: The Mechanics of Serial and Parallel Manipulators**; 1.ed.; New York: John Wiley & Sons; 1999; 520p.
- [30] Spong, M. W.; Hutchinson, S.; Vidyasagar, M.; **Robot Modeling and Control**; 1.ed.; New York: John Wiley & Sons; 2006; 496p.

Apêndice A – Programas dos microcontroladores

Abaixo é apresentada a rotina implementada no microcontrolador PIC18F2431 com o compilador C CCS. Apenas o programa da junta 1 é apresentado. As rotinas das outras juntas são iguais, somente alguns parâmetros do controle mudam. Estes parâmetros podem ser encontrados na Tabela 6.

```
// Controle de Motor de Passo
// Compilador C CCS
// PD com aceleração e Velocidade Limitada
// Malha Fechada com Encoder
// Leitura do Encoder com QEI
// Entrada: STEP/DIR
// Saída: STEP/DIR
// Hardware: PIC18F2431
// Autor: William Cardozo
// Data: 8/12/11
//
// Junta 1
// Motor em meio passo
// Encoder: 2000inc/volta
// Redutor do motor: 20:1
// Multiplicador do Encoder: 3:1

#include "18F2431.h"
#use delay(clock=2000000)
#use fast_io(a)
//#use fast_io(b)
//#use fast_io(c)
#fuses HS,NOLVP,NOWDT
//#fuses NOMCLR , HS, NOWDT, NOPUT, NOBROWNOUT, NOLVP, PWMPIN
//#use rs232 (baud = 19200, xmit = PIN_C6, rcv = PIN_C7)

// Implementa registradores (SFRs)
#byte TRISC = 0xf3
#byte TRISB = 0xf93
```

```

#byte TRISA    = 0xf92
#byte PORTC    = 0xf82
#byte PORTB    = 0xf81
#byte PORTA    = 0xf80

//Definição de Pinos de Saída
#bit dir_out   = portc.6
#bit step_out  = portc.7

//Definição de Pinos de Entrada
#bit dir_in    = portc.4
#bit step_in   = portc.3

//Definição dos Estados de Saída
#define PULSE_ON  0
#define PULSE_OFF 1
#define DIR_ON    1
#define DIR_OFF   0

//configuracoes QEI
#byte QEICON      = 0xFB6    // Registrador do QEI
#byte DFLTCON     = 0xF60    // Registrador do Filtro
#byte POSCNTH     = 0xF67    // Registrador do byte Alto da
posição
#byte POSCNTL     = 0xF66    // Registrador do byte Baixo da
posição
int16 POSCNT;      // Posição atual
#byte POSCNT      = 0xF66    // Registrador do byte Baixo da
posição
int16 MAXCNT;
#byte MAXCNT      = 0xF64
#byte MAXCNTH     = 0xF65
#byte MAXCNTL     = 0xF64

//Variáveis e Constantes
signed long int pd=0;      //Saída PD
signed long int pd_max=0;  //Saída PD máxima dentro do
limite de aceleração
signed long int pd_min=0;  //Saída PD mínima dentro do
limite de desaceleração
signed long int erro=0;    //erro

```

```

signed long int erro_ult=0; // utlimo erro
signed long int dv=0; // erro_atual-erro_utlimo
long int pos_d = 10000; // posição desejada inicial
(arbitrado como 0.5*MAXCNT)
long int alfa_max = 50; // aceleração máxima*0.01
#define vel_d_min 25 // Velocidade mínima imposta ao
motor
#define vel_d_max 1000 // Velocidade máxima imposta ao
motor

int16 vel_d; // usado para calcular o
período
int16 i_int_vel; // contador de interrupção
int16 const_t_vel; // quando o contador for maior
que isto->calc vel
int1 dir_change=0; // flag registra se direção
mudou
int1 RUN=0; // flag registra se motor deve
gitar
int inc_pos=0; // incremento na posição
desejada (0->-1 e 1->1)

//PD
int8 kp=13;
// Junta2:kp=10 Junta3:kp=15 Junta4:kp=6 Junta5e6:kp=10
int8 kd=2; // kd/dt
//PD se erro pequeno
int8 kp_c=2;

// Período do Timer 1
//  $T = (65536 - \text{XXXX}) / (20M / 4 / 1)$ 
int16 const_t1;
#define CONST_T1_MAX 64286 // Resulta em 250us
#define CONST_T1_MIN 15536 // Resulta em 10ms

#priority int_ext1,timer1
// Interrupção timer1
// gerar pulsos com frequencia variável
// loop de controle a cada 0.01s
#int_timer1
void leitura()

```

```

{
    // Se o controle esta em RUN -> gerar pulso
    if((step_out==PULSE_ON))
        step_out = PULSE_OFF;
    else
    {
        if( RUN )
            step_out = PULSE_ON;
    }

    if(i_int_vel>=const_t_vel) // a cada 0.01s calcular
    velocidade de saída
    {
        if(dir_in==1)
        {
            pos_d = pos_d+get_timer0(); // soma pulsos contados
            da pos_desejada
            set_timer0(0);
        }
        else
        {
            pos_d = pos_d-get_timer0(); // subtrai pulsos
            contados da pos_desejada
            set_timer0(0);
        }

        if(RUN)
            Vel = pd;
        else
            Vel=0;

        erro_ult = erro; // ultimo erro
        erro = pos_d-POSCNT; // erro atual
        dv = erro-erro_ult; // diferenca

        pd=erro*kp+kd*dv; // cáclulo PD

        // Se a junta estiver próxima a posição desejada:
        // Executa PD com menores const. por causa do backlash

        if((erro<=10)&&(erro>=-10))

```



```

    pd=erro*kp_c; //+kd_c*dv;

    //Admite |erro|<=1
    if((erro<=1)&&(erro>=-1))
        pd=0; //+kd_c*dv;

    pd_max=Vel+alfa_max; // calcula vel. maxima dentro do
limite de acel.
    pd_min=Vel-alfa_max; // calcula vel. minima dentro do
limite de acel.

    if(pd<pd_min)
        pd=pd_min;

    if(pd>pd_max)
        pd=pd_max;

    if((pd<0)&&(pd>-vel_d_min))
        pd=-vel_d_min;

    if((pd>0)&&(pd<vel_d_min))
        pd=vel_d_min;

    if(pd<-vel_d_max)
        pd=-vel_d_max;

    if(pd>vel_d_max)
        pd=vel_d_max;

    // determina se o pulso de direção da saída
    // deve estar em nível alto ou baixo
    if(pd>=0)
    {
        if(dir_out==DIR_ON)
        {
            dir_change=1;
            vel_d=0;
        }
        else
        {
            dir_change=0;

```

```

        vel_d = abs(pd);
    }
    dir_out = DIR_OFF;
}

if(pd<0)
{
    if(dir_out==DIR_OFF)
    {
        dir_change=1;
        vel_d=0;
    }
    else
    {
        dir_change=0;
        vel_d = abs(pd);
    }
    dir_out = DIR_ON;
}

// se vel. na saída é nula: RUN=0 faz não gerar pulsos
// mas timer deve continuar a gerar interrupções para
// o loop de controle
if(vel_d==0)
{
    const_t1=CONST_T1_MIN;
    const_t_vel=10; // =100ms
    RUN=0;
}
else
{
    // calcula quantas vez periodo de timer1 é maior que
    // período de controle
    const_t_vel=4*vel_d;
    const_t_vel=const_t_vel/100;

    // calcular valor do timer1 para frequencia desejada
    const_t1=vel_d;
    const_t1=50000/const_t1;
    const_t1=const_t1*25+1;
    const_t1=65536-const_t1;
}

```

```

        RUN=1;
    }
    i_int_vel=0;
}
i_int_vel=i_int_vel+1;
set_timer1(const_t1+get_timer1());
}

// Interrupção para computar os pulsos quando a entrada dir
muda de estado
#int_ext1
void isrext()
{
    inc_pos=dir_in;
    if(dir_in==0) // logo estava em 1
    {
        pos_d = pos_d+get_timer0();
        set_timer0(0);
        ext_int_edge(1,L_TO_H);
    }
    else // logo estava em 0
    {
        pos_d = pos_d-get_timer0();
        set_timer0(0);
        ext_int_edge(1,H_TO_L);
    }
}

void main()
{
    set_tris_b(0b00000000);
    set_tris_c(0b00011000);

    dir_out=DIR_OFF;
    step_out=PULSE_OFF;

    const_t1=CONST_T1_MAX;
    const_t_vel=0;
    i_int_vel=0; // contador interrupção
    RUN=0;
}

```

```

// Configurações da QEI
setup_timer_5(T5_INTERNAL|T5_DIV_BY_1);
QEICON=0b10111000;
//0b10111000;VelocDisable,NoError,ForwardDir,4xUpdate,IndxResetsPos,1:1
DFLTCON=0b01111011;
//NoiseFilterTCKI5,QEB,QEA,INDX,Noise1:128
MAXCNT = 20000; // maximo valor da medida do encoder
POSCNT = pos_d; // inicialmente posição atual é a
demanda

// configura e inicializa interrupções
setup_timer_1( T1_INTERNAL | T1_DIV_BY_1 );
set_timer1(const_t1);

setup_timer_0( RTCC_EXT_L_TO_H | RTCC_DIV_1);
set_timer0(0);

if(dir_in==0)
    ext_int_edge(1,L_TO_H);
else
    ext_int_edge(1,H_TO_L);

enable_interrupts(GLOBAL);
enable_interrupts (INT_IC2QEI);
enable_interrupts (int_timer1);
enable_interrupts (int_ext1);

while(true)
{
    //loop infinito
}
}

```

Apêndice B – Códigos do MATLAB

Função para simulação do manipulador

```
function
[T1, T2, T3, T4, T5, T6, qe, r_01, r_02, r_03, r_04, r_05, r_06, x_6, y_6, z_6] =
Manipulador_dinamica(Theta1, DTheta1, D2Theta1, Theta2, DTheta2, D2Theta2,
Theta3, DTheta3, D2Theta3, Theta4, DTheta4, D2Theta4, Theta5, DTheta5,
D2Theta5, Theta6, DTheta6, D2Theta6)
% Parametro de DH
alfa1=pi/2;
alfa2=0;
alfa3=pi/2;
alfa4=-pi/2;
alfa5=pi/2;
alfa6=0;

a1=0.1;
a2=0.25;
a3=0;
a4=0;
a5=0;
a6=0;

d1=0;
d2=0;
d3=0;
d4=0.16;
d5=0;
d6=0.05;

% Matrizes de Transformação
A_01=[[cos(Theta1), -
sin(Theta1)*cos(alfa1), sin(Theta1)*sin(alfa1), a1*cos(Theta1)]; [sin
(Theta1), cos(Theta1)*cos(alfa1), -
cos(Theta1)*sin(alfa1), a1*sin(Theta1)]; [0, sin(alfa1), cos(alfa1), d1
]; [0, 0, 0, 1]];
A_12=[[cos(Theta2), -
sin(Theta2)*cos(alfa2), sin(Theta2)*sin(alfa2), a2*cos(Theta2)]; [sin
(Theta2), cos(Theta2)*cos(alfa2), -
cos(Theta2)*sin(alfa2), a2*sin(Theta2)]; [0, sin(alfa2), cos(alfa2), d2
]; [0, 0, 0, 1]];
A_23=[[cos(Theta3), -
sin(Theta3)*cos(alfa3), sin(Theta3)*sin(alfa3), a3*cos(Theta3)]; [sin
(Theta3), cos(Theta3)*cos(alfa3), -
cos(Theta3)*sin(alfa3), a3*sin(Theta3)]; [0, sin(alfa3), cos(alfa3), d3
]; [0, 0, 0, 1]];
A_34=[[cos(Theta4), -
sin(Theta4)*cos(alfa4), sin(Theta4)*sin(alfa4), a4*cos(Theta4)]; [sin
(Theta4), cos(Theta4)*cos(alfa4), -
cos(Theta4)*sin(alfa4), a4*sin(Theta4)]; [0, sin(alfa4), cos(alfa4), d4
]; [0, 0, 0, 1]];
```

```

A_45=[ [cos(Theta5), -
sin(Theta5)*cos(alfa5), sin(Theta5)*sin(alfa5), a5*cos(Theta5)]; [sin
(Theta5), cos(Theta5)*cos(alfa5), -
cos(Theta5)*sin(alfa5), a5*sin(Theta5)]; [0, sin(alfa5), cos(alfa5), d5
]; [0, 0, 0, 1]];
A_56=[ [cos(Theta6), -
sin(Theta6)*cos(alfa6), sin(Theta6)*sin(alfa6), a6*cos(Theta6)]; [sin
(Theta6), cos(Theta6)*cos(alfa6), -
cos(Theta6)*sin(alfa6), a6*sin(Theta6)]; [0, sin(alfa6), cos(alfa6), d6
]; [0, 0, 0, 1]];

A_02=A_01*A_12;
A_03=A_01*A_12*A_23;
A_04=A_01*A_12*A_23*A_34;
A_05=A_01*A_12*A_23*A_34*A_45;
A_06=A_01*A_12*A_23*A_34*A_45*A_56;
A_13=A_12*A_23;
A_14=A_12*A_23*A_34;
A_15=A_12*A_23*A_34*A_45;
A_16=A_12*A_23*A_34*A_45*A_56;
A_24=A_23*A_34;
A_25=A_23*A_34*A_45;
A_26=A_23*A_34*A_45*A_56;
A_35=A_34*A_45;
A_36=A_34*A_45*A_56;
A_46=A_45*A_56;

qe=[A_06(1,4);A_06(2,4);A_06(3,4)]; % Posição da Extremidade

% Matrizes de Rotação
R_01=[ [A_01(1,1), A_01(1,2), A_01(1,3)]; [A_01(2,1), A_01(2,2), A_01(2,
3)]; [A_01(3,1), A_01(3,2), A_01(3,3)]];
R_12=[ [A_12(1,1), A_12(1,2), A_12(1,3)]; [A_12(2,1), A_12(2,2), A_12(2,
3)]; [A_12(3,1), A_12(3,2), A_12(3,3)]];
R_23=[ [A_23(1,1), A_23(1,2), A_23(1,3)]; [A_23(2,1), A_23(2,2), A_23(2,
3)]; [A_23(3,1), A_23(3,2), A_23(3,3)]];
R_34=[ [A_34(1,1), A_34(1,2), A_34(1,3)]; [A_34(2,1), A_34(2,2), A_34(2,
3)]; [A_34(3,1), A_34(3,2), A_34(3,3)]];
R_45=[ [A_45(1,1), A_45(1,2), A_45(1,3)]; [A_45(2,1), A_45(2,2), A_45(2,
3)]; [A_45(3,1), A_45(3,2), A_45(3,3)]];
R_56=[ [A_56(1,1), A_56(1,2), A_56(1,3)]; [A_56(2,1), A_56(2,2), A_56(2,
3)]; [A_56(3,1), A_56(3,2), A_56(3,3)]];

R_02=[ [A_02(1,1), A_02(1,2), A_02(1,3)]; [A_02(2,1), A_02(2,2), A_02(2,
3)]; [A_02(3,1), A_02(3,2), A_02(3,3)]];
R_03=[ [A_03(1,1), A_03(1,2), A_03(1,3)]; [A_03(2,1), A_03(2,2), A_03(2,
3)]; [A_03(3,1), A_03(3,2), A_03(3,3)]];
R_04=[ [A_04(1,1), A_04(1,2), A_04(1,3)]; [A_04(2,1), A_04(2,2), A_04(2,
3)]; [A_04(3,1), A_04(3,2), A_04(3,3)]];
R_05=[ [A_05(1,1), A_05(1,2), A_05(1,3)]; [A_05(2,1), A_05(2,2), A_05(2,
3)]; [A_05(3,1), A_05(3,2), A_05(3,3)]];
R_06=[ [A_06(1,1), A_06(1,2), A_06(1,3)]; [A_06(2,1), A_06(2,2), A_06(2,
3)]; [A_06(3,1), A_06(3,2), A_06(3,3)]];

% Posição de cada junta em relação ao sistema da base
r_01=[A_01(1,4);A_01(2,4);A_01(3,4)];
r_02=[A_02(1,4);A_02(2,4);A_02(3,4)];
r_03=[A_03(1,4);A_03(2,4);A_03(3,4)];
r_04=[A_04(1,4);A_04(2,4);A_04(3,4)];
r_05=[A_05(1,4);A_05(2,4);A_05(3,4)];

```

```

r_06=[A_06(1,4);A_06(2,4);A_06(3,4)];

% Posição de cada junta a frente em relação ao sistema da junta 1
r_12=[A_12(1,4);A_12(2,4);A_12(3,4)];
r_13=[A_13(1,4);A_13(2,4);A_13(3,4)];
r_14=[A_14(1,4);A_14(2,4);A_14(3,4)];
r_15=[A_15(1,4);A_15(2,4);A_15(3,4)];
r_16=[A_16(1,4);A_16(2,4);A_16(3,4)];

% Posição de cada junta a frente em relação ao sistema da junta 1
r_23=[A_23(1,4);A_23(2,4);A_23(3,4)];
r_24=[A_24(1,4);A_24(2,4);A_24(3,4)];
r_25=[A_25(1,4);A_25(2,4);A_25(3,4)];
r_26=[A_26(1,4);A_26(2,4);A_26(3,4)];

% Posição de cada junta a frente em relação ao sistema da junta 1
r_34=[A_34(1,4);A_34(2,4);A_34(3,4)];
r_35=[A_35(1,4);A_35(2,4);A_35(3,4)];
r_36=[A_36(1,4);A_36(2,4);A_36(3,4)];

% Posição de cada junta a frente em relação ao sistema da junta 1
r_45=[A_45(1,4);A_45(2,4);A_45(3,4)];
r_46=[A_46(1,4);A_46(2,4);A_46(3,4)];

% Posição de cada junta a frente em relação ao sistema da junta 1
r_56=[A_56(1,4);A_56(2,4);A_56(3,4)];

% Eixos z de cada junta
z_0=[0;0;1];
z_1=[A_01(1,3);A_01(2,3);A_01(3,3)];
z_2=[A_02(1,3);A_02(2,3);A_02(3,3)];
z_3=[A_03(1,3);A_03(2,3);A_03(3,3)];
z_4=[A_04(1,3);A_04(2,3);A_04(3,3)];
z_5=[A_05(1,3);A_05(2,3);A_05(3,3)];
z_6=[A_06(1,3);A_06(2,3);A_06(3,3)];

% Eixos x e y da extremidade (para gráfico)
x_6=[A_06(1,1);A_06(2,1);A_06(3,1)];
y_6=[A_06(1,2);A_06(2,2);A_06(3,2)];

% Massa dos elos
m1=12.107;
m2=21.699;
m3=3.686;
m4=4.019;
m5=0.203;
m6=0.369;

% Posição dos centros de massa das juntas em relação
% ao sistema fixo a junta
rc11=[-0.027228;-0.047041;0.04162];
rc22=[-0.3384;0.000384;-0.132453];
rc33=[0.001931;-0.031655;-0.113306];
rc44=[-0.000133;0.119374;0.000103];
rc55=[0;0.000257;0.004616];
rc66=[0;0;0.012196];

% Matriz de Inércia do CM usando sistema fixa de cada junta

```

```

Ig11=[[0.1314,-0.0178,0.0143];[-0.0178,0.1314,-0.025];[0.0143,-
0.025,0.0727]];
Ig22=[[0.06258,-0.0028,-0.005];[-0.0028,0.5190,-0.0002];[-0.0050,-
0.0002,0.5418]];
Ig33=[[0.0222,0.0002,0];[0.0002,0.01959,0.0058];[0,0.0058,0.0059]]
;
Ig44=[[0.02809,0,0];[0,0.0044,0];[0,0,0.026]];
Ig55=[[0.0001,0,0];[0,0.00012,0];[0,0,0.00011]];
Ig66=[[0.00011,0,0];[0,0.00011,0];[0,0,0.00017]];

% o vetor que define a posição do centro de massa do elo i
% em relação a origem do elo j-1
% escrito no sistema de coordenadas da base.
p0c1=r_01+R_01*rc11;
p0c2=r_02+R_01*R_12*rc22;
p0c3=r_03+R_01*R_12*R_23*rc33;
p0c4=r_04+R_01*R_12*R_23*R_34*rc44;
p0c5=r_05+R_01*R_12*R_23*R_34*R_45*rc55;
p0c6=r_06+R_01*R_12*R_23*R_34*R_45*R_56*rc66;

p1c1=[0;0;0];
p1c2=r_12+R_12*rc22;
p1c3=r_13+R_12*R_23*rc33;
p1c4=r_14+R_12*R_23*R_34*rc44;
p1c5=r_15+R_12*R_23*R_34*R_45*rc55;
p1c6=r_16+R_12*R_23*R_34*R_45*R_56*rc66;

p2c1=[0;0;0];
p2c2=[0;0;0];
p2c3=r_23+R_23*rc33;
p2c4=r_24+R_23*R_34*rc44;
p2c5=r_25+R_23*R_34*R_45*rc55;
p2c6=r_26+R_23*R_34*R_45*R_56*rc66;

p3c1=[0;0;0];
p3c2=[0;0;0];
p3c3=[0;0;0];
p3c4=r_34+R_34*rc44;
p3c5=r_35+R_34*R_45*rc55;
p3c6=r_36+R_34*R_45*R_56*rc66;

p4c1=[0;0;0];
p4c2=[0;0;0];
p4c3=[0;0;0];
p4c4=[0;0;0];
p4c5=r_45+R_45*rc55;
p4c6=r_46+R_45*R_56*rc66;

p5c1=[0;0;0];
p5c2=[0;0;0];
p5c3=[0;0;0];
p5c4=[0;0;0];
p5c5=[0;0;0];
p5c6=r_56+R_56*rc66;

% sub-matriz Jacobiana associada à velocidade linear do centro de
massa
Jv1=[cross(z_0,p0c1),[0;0;0]          , [0;0;0]          , [0;0;0]
, [0;0;0]          , [0;0;0]];

```



```

Jv2=[cross(z_0,p0c2),cross(z_1,plc2),[0;0;0]      , [0;0;0]
, [0;0;0]      , [0;0;0]];
Jv3=[cross(z_0,p0c3),cross(z_1,plc3),cross(z_2,p2c3), [0;0;0]
, [0;0;0]      , [0;0;0]];
Jv4=[cross(z_0,p0c4),cross(z_1,plc4),cross(z_2,p2c4),cross(z_3,p3c
4), [0;0;0]      , [0;0;0]];
Jv5=[cross(z_0,p0c5),cross(z_1,plc5),cross(z_2,p2c5),cross(z_3,p3c
5),cross(z_4,p4c5), [0;0;0]];
Jv6=[cross(z_0,p0c6),cross(z_1,plc6),cross(z_2,p2c6),cross(z_3,p3c
6),cross(z_4,p4c6),cross(z_5,p5c6)];

```

```

% sub-matriz Jacobiana associada à velocidade angular

```

```

Jw1=[z_0,[0;0;0],[0;0;0],[0;0;0],[0;0;0],[0;0;0];
Jw2=[z_0,z_1      , [0;0;0],[0;0;0],[0;0;0],[0;0;0]];
Jw3=[z_0,z_1      , z_2      , [0;0;0],[0;0;0],[0;0;0]];
Jw4=[z_0,z_1      , z_2      , z_3      , [0;0;0],[0;0;0]];
Jw5=[z_0,z_1      , z_2      , z_3      , z_4      , [0;0;0]];
Jw6=[z_0,z_1      , z_2      , z_3      , z_4      , z_5];

```

```

% Rotação das matrizes de inércia dos elos para sistema fixo

```

```

Ig1=R_01*Ig11*R_01';
Ig2=R_02*Ig22*R_02';
Ig3=R_03*Ig33*R_03';
Ig4=R_04*Ig44*R_04';
Ig5=R_05*Ig55*R_05';
Ig6=R_06*Ig66*R_06';

```

```

g=[0;0;-9.81]; % vetor gravidade

```

```

% Matriz de Inércia do Manipulador

```

```

M=Jv1'*m1*Jv1+Jw1'*Ig1*Jw1 + Jv2'*m2*Jv2+Jw2'*Ig2*Jw2 +
Jv3'*m3*Jv3+Jw3'*Ig3*Jw3 + Jv4'*m4*Jv4+Jw4'*Ig4*Jw4 +
Jv5'*m5*Jv5+Jw5'*Ig5*Jw5 + Jv6'*m6*Jv6+Jw6'*Ig6*Jw6;

```

```

% Torques gravitacionais

```

```

G1=-m1*g'*Jv1(:,1)-m2*g'*Jv2(:,1)-m3*g'*Jv3(:,1)-m4*g'*Jv4(:,1)-
m5*g'*Jv5(:,1)-m6*g'*Jv6(:,1);
G2=-m1*g'*Jv1(:,2)-m2*g'*Jv2(:,2)-m3*g'*Jv3(:,2)-m4*g'*Jv4(:,2)-
m5*g'*Jv5(:,2)-m6*g'*Jv6(:,2);
G3=-m1*g'*Jv1(:,3)-m2*g'*Jv2(:,3)-m3*g'*Jv3(:,3)-m4*g'*Jv4(:,3)-
m5*g'*Jv5(:,3)-m6*g'*Jv6(:,3);
G4=-m1*g'*Jv1(:,4)-m2*g'*Jv2(:,4)-m3*g'*Jv3(:,4)-m4*g'*Jv4(:,4)-
m5*g'*Jv5(:,4)-m6*g'*Jv6(:,4);
G5=-m1*g'*Jv1(:,5)-m2*g'*Jv2(:,5)-m3*g'*Jv3(:,5)-m4*g'*Jv4(:,5)-
m5*g'*Jv5(:,5)-m6*g'*Jv6(:,5);
G6=-m1*g'*Jv1(:,6)-m2*g'*Jv2(:,6)-m3*g'*Jv3(:,6)-m4*g'*Jv4(:,6)-
m5*g'*Jv5(:,6)-m6*g'*Jv6(:,6);

```

```

% Torque devido o momento de inércia

```

```

sum_M1j=M(1,1)*D2Theta1+M(1,2)*D2Theta2+M(1,3)*D2Theta3+M(1,4)*D2T
heta4+M(1,5)*D2Theta5+M(1,6)*D2Theta6;
sum_M2j=M(2,1)*D2Theta1+M(2,2)*D2Theta2+M(2,3)*D2Theta3+M(2,4)*D2T
heta4+M(2,5)*D2Theta5+M(2,6)*D2Theta6;
sum_M3j=M(3,1)*D2Theta1+M(3,2)*D2Theta2+M(3,3)*D2Theta3+M(3,4)*D2T
heta4+M(3,5)*D2Theta5+M(3,6)*D2Theta6;
sum_M4j=M(4,1)*D2Theta1+M(4,2)*D2Theta2+M(4,3)*D2Theta3+M(4,4)*D2T
heta4+M(4,5)*D2Theta5+M(4,6)*D2Theta6;
sum_M5j=M(5,1)*D2Theta1+M(5,2)*D2Theta2+M(5,3)*D2Theta3+M(5,4)*D2T
heta4+M(5,5)*D2Theta5+M(5,6)*D2Theta6;

```

```

sum_M6j=M(6,1)*D2Theta1+M(6,2)*D2Theta2+M(6,3)*D2Theta3+M(6,4)*D2T
heta4+M(6,5)*D2Theta5+M(6,6)*D2Theta6;

q=[Theta1,Theta2,Theta3,Theta4,Theta5,Theta6]; % Vetor posição das
juntas
dq=[DTheta1,DTheta2,DTheta3,DTheta4,DTheta5,DTheta6]; % Vetor
velocidade das juntas

% Cálculo numérico da derivada da matriz de inércia do manipulador

delta_theta=0.01; % incremento ininitesimal para cálculo da
derivada aprox.

dM=zeros(6,6,6); % Inicializa matriz

% Matriz com inc. em Theta1
dM(:,:,1)=M_gen(Theta1+delta_theta,Theta2,Theta3,Theta4,Theta5,The
ta6);
% Matriz com inc. em Theta2
dM(:,:,2)=M_gen(Theta1,Theta2+delta_theta,Theta3,Theta4,Theta5,The
ta6);
% Matriz com inc. em Theta3
dM(:,:,3)=M_gen(Theta1,Theta2,Theta3+delta_theta,Theta4,Theta5,The
ta6);
% Matriz com inc. em Theta4
dM(:,:,4)=M_gen(Theta1,Theta2,Theta3,Theta4+delta_theta,Theta5,The
ta6);
% Matriz com inc. em Theta5
dM(:,:,5)=M_gen(Theta1,Theta2,Theta3,Theta4,Theta5+delta_theta,The
ta6);
% Matriz com inc. em Theta6
dM(:,:,6)=M_gen(Theta1,Theta2,Theta3,Theta4,Theta5,Theta6+delta_th
eta);

V1=0;
V2=0;
V3=0;
V4=0;
V5=0;
V6=0;

% Cálculo Centrífugos e Coriolis
for j=1:6
    for k=1:6
        V1=V1+((dM(1,j,k)-M(1,j))/delta_theta-0.5*(dM(j,k,1)-
M(j,k))/delta_theta)*dq(j)*dq(k);
        V2=V2+((dM(2,j,k)-M(2,j))/delta_theta-0.5*(dM(j,k,2)-
M(j,k))/delta_theta)*dq(j)*dq(k);
        V3=V3+((dM(3,j,k)-M(3,j))/delta_theta-0.5*(dM(j,k,3)-
M(j,k))/delta_theta)*dq(j)*dq(k);
        V4=V4+((dM(4,j,k)-M(4,j))/delta_theta-0.5*(dM(j,k,4)-
M(j,k))/delta_theta)*dq(j)*dq(k);
        V5=V5+((dM(5,j,k)-M(5,j))/delta_theta-0.5*(dM(j,k,5)-
M(j,k))/delta_theta)*dq(j)*dq(k);
        V6=V6+((dM(6,j,k)-M(6,j))/delta_theta-0.5*(dM(j,k,6)-
M(j,k))/delta_theta)*dq(j)*dq(k);
    end
end

% Calcula Torque Total

```

```
T1=sum_M1j+V1+G1;
T2=sum_M2j+V2+G2;
T3=sum_M3j+V3+G3;
T4=sum_M4j+V4+G4;
T5=sum_M5j+V5+G5;
T6=sum_M6j+V6+G6;
```

Função para cálculo da matriz de inércia

```
function [ M ] = M_gen( Theta1, Theta2, Theta3, Theta4, Theta5,
Theta6 )
% Parametro de DH
alfa1=pi/2;
alfa2=0;
alfa3=pi/2;
alfa4=-pi/2;
alfa5=pi/2;
alfa6=0;

a1=0.1;
a2=0.25;
a3=0;
a4=0;
a5=0;
a6=0;

d1=0;
d2=0;
d3=0;
d4=0.16;
d5=0;
d6=0.05;

% Matrizes de Transformação
A_01=[[cos(Theta1), -
sin(Theta1)*cos(alfa1), sin(Theta1)*sin(alfa1), a1*cos(Theta1)]; [sin
(Theta1), cos(Theta1)*cos(alfa1), -
cos(Theta1)*sin(alfa1), a1*sin(Theta1)]; [0, sin(alfa1), cos(alfa1), d1
]; [0, 0, 0, 1]];
A_12=[[cos(Theta2), -
sin(Theta2)*cos(alfa2), sin(Theta2)*sin(alfa2), a2*cos(Theta2)]; [sin
(Theta2), cos(Theta2)*cos(alfa2), -
cos(Theta2)*sin(alfa2), a2*sin(Theta2)]; [0, sin(alfa2), cos(alfa2), d2
]; [0, 0, 0, 1]];
A_23=[[cos(Theta3), -
sin(Theta3)*cos(alfa3), sin(Theta3)*sin(alfa3), a3*cos(Theta3)]; [sin
(Theta3), cos(Theta3)*cos(alfa3), -
cos(Theta3)*sin(alfa3), a3*sin(Theta3)]; [0, sin(alfa3), cos(alfa3), d3
]; [0, 0, 0, 1]];
A_34=[[cos(Theta4), -
sin(Theta4)*cos(alfa4), sin(Theta4)*sin(alfa4), a4*cos(Theta4)]; [sin
(Theta4), cos(Theta4)*cos(alfa4), -
cos(Theta4)*sin(alfa4), a4*sin(Theta4)]; [0, sin(alfa4), cos(alfa4), d4
]; [0, 0, 0, 1]];
A_45=[[cos(Theta5), -
sin(Theta5)*cos(alfa5), sin(Theta5)*sin(alfa5), a5*cos(Theta5)]; [sin
(Theta5), cos(Theta5)*cos(alfa5), -
cos(Theta5)*sin(alfa5), a5*sin(Theta5)]; [0, sin(alfa5), cos(alfa5), d5
]; [0, 0, 0, 1]];
```

```

A_56=[ [cos(Theta6), -
sin(Theta6)*cos(alfa6), sin(Theta6)*sin(alfa6), a6*cos(Theta6)]; [sin
(Theta6), cos(Theta6)*cos(alfa6), -
cos(Theta6)*sin(alfa6), a6*sin(Theta6)]; [0, sin(alfa6), cos(alfa6), d6
]; [0,0,0,1]];

A_02=A_01*A_12;
A_03=A_01*A_12*A_23;
A_04=A_01*A_12*A_23*A_34;
A_05=A_01*A_12*A_23*A_34*A_45;
A_06=A_01*A_12*A_23*A_34*A_45*A_56;
A_13=A_12*A_23;
A_14=A_12*A_23*A_34;
A_15=A_12*A_23*A_34*A_45;
A_16=A_12*A_23*A_34*A_45*A_56;
A_24=A_23*A_34;
A_25=A_23*A_34*A_45;
A_26=A_23*A_34*A_45*A_56;
A_35=A_34*A_45;
A_36=A_34*A_45*A_56;
A_46=A_45*A_56;

% Matrizes de Rotação
R_01=[ [A_01(1,1), A_01(1,2), A_01(1,3)]; [A_01(2,1), A_01(2,2), A_01(2,
3)]; [A_01(3,1), A_01(3,2), A_01(3,3)]];
R_12=[ [A_12(1,1), A_12(1,2), A_12(1,3)]; [A_12(2,1), A_12(2,2), A_12(2,
3)]; [A_12(3,1), A_12(3,2), A_12(3,3)]];
R_23=[ [A_23(1,1), A_23(1,2), A_23(1,3)]; [A_23(2,1), A_23(2,2), A_23(2,
3)]; [A_23(3,1), A_23(3,2), A_23(3,3)]];
R_34=[ [A_34(1,1), A_34(1,2), A_34(1,3)]; [A_34(2,1), A_34(2,2), A_34(2,
3)]; [A_34(3,1), A_34(3,2), A_34(3,3)]];
R_45=[ [A_45(1,1), A_45(1,2), A_45(1,3)]; [A_45(2,1), A_45(2,2), A_45(2,
3)]; [A_45(3,1), A_45(3,2), A_45(3,3)]];
R_56=[ [A_56(1,1), A_56(1,2), A_56(1,3)]; [A_56(2,1), A_56(2,2), A_56(2,
3)]; [A_56(3,1), A_56(3,2), A_56(3,3)]];

R_02=[ [A_02(1,1), A_02(1,2), A_02(1,3)]; [A_02(2,1), A_02(2,2), A_02(2,
3)]; [A_02(3,1), A_02(3,2), A_02(3,3)]];
R_03=[ [A_03(1,1), A_03(1,2), A_03(1,3)]; [A_03(2,1), A_03(2,2), A_03(2,
3)]; [A_03(3,1), A_03(3,2), A_03(3,3)]];
R_04=[ [A_04(1,1), A_04(1,2), A_04(1,3)]; [A_04(2,1), A_04(2,2), A_04(2,
3)]; [A_04(3,1), A_04(3,2), A_04(3,3)]];
R_05=[ [A_05(1,1), A_05(1,2), A_05(1,3)]; [A_05(2,1), A_05(2,2), A_05(2,
3)]; [A_05(3,1), A_05(3,2), A_05(3,3)]];
R_06=[ [A_06(1,1), A_06(1,2), A_06(1,3)]; [A_06(2,1), A_06(2,2), A_06(2,
3)]; [A_06(3,1), A_06(3,2), A_06(3,3)]];

% Posição de cada junta
r_01=[A_01(1,4); A_01(2,4); A_01(3,4)];
r_02=[A_02(1,4); A_02(2,4); A_02(3,4)];
r_03=[A_03(1,4); A_03(2,4); A_03(3,4)];
r_04=[A_04(1,4); A_04(2,4); A_04(3,4)];
r_05=[A_05(1,4); A_05(2,4); A_05(3,4)];
r_06=[A_06(1,4); A_06(2,4); A_06(3,4)];
r_12=[A_12(1,4); A_12(2,4); A_12(3,4)];
r_13=[A_13(1,4); A_13(2,4); A_13(3,4)];
r_14=[A_14(1,4); A_14(2,4); A_14(3,4)];
r_15=[A_15(1,4); A_15(2,4); A_15(3,4)];
r_16=[A_16(1,4); A_16(2,4); A_16(3,4)];
r_23=[A_23(1,4); A_23(2,4); A_23(3,4)];
r_24=[A_24(1,4); A_24(2,4); A_24(3,4)];

```

```

r_25=[A_25(1,4);A_25(2,4);A_25(3,4)];
r_26=[A_26(1,4);A_26(2,4);A_26(3,4)];
r_34=[A_34(1,4);A_34(2,4);A_34(3,4)];
r_35=[A_35(1,4);A_35(2,4);A_35(3,4)];
r_36=[A_36(1,4);A_36(2,4);A_36(3,4)];
r_45=[A_45(1,4);A_45(2,4);A_45(3,4)];
r_46=[A_46(1,4);A_46(2,4);A_46(3,4)];
r_56=[A_56(1,4);A_56(2,4);A_56(3,4)];

% Eixos z de cada junta
z_0=[0;0;1];
z_1=[A_01(1,3);A_01(2,3);A_01(3,3)];
z_2=[A_02(1,3);A_02(2,3);A_02(3,3)];
z_3=[A_03(1,3);A_03(2,3);A_03(3,3)];
z_4=[A_04(1,3);A_04(2,3);A_04(3,3)];
z_5=[A_05(1,3);A_05(2,3);A_05(3,3)];
z_6=[A_06(1,3);A_06(2,3);A_06(3,3)];

% Massa dos elos
m1=12.107;
m2=21.699;
m3=3.686;
m4=4.019;
m5=0.203;
m6=0.369;

% Posição dos centros de massa das juntas em relação
% ao sistema fixo a junta
rc11=[-0.027228;-0.047041;0.04162];
rc22=[-0.3384;0.000384;-0.132453];
rc33=[0.001931;-0.031655;-0.113306];
rc44=[-0.000133;0.119374;0.000103];
rc55=[0;0.000257;0.004616];
rc66=[0;0;0.012196];

% Matriz de Inércia do CM usando sistema fixa de cada junta
Ig11=[[0.1314,-0.0178,0.0143];[-0.0178,0.1314,-0.025];[0.0143,-
0.025,0.0727]];
Ig22=[[0.06258,-0.0028,-0.005];[-0.0028,0.5190,-0.0002];[-0.0050,-
0.0002,0.5418]];
Ig33=[[0.0222,0.0002,0];[0.0002,0.01959,0.0058];[0,0.0058,0.0059]]
;
Ig44=[[0.02809,0,0];[0,0.0044,0];[0,0,0.026]];
Ig55=[[0.0001,0,0];[0,0.00012,0];[0,0,0.00011]];
Ig66=[[0.00011,0,0];[0,0.00011,0];[0,0,0.00017]];

% o vetor que define a posição do centro de massa do elo i
% em relação a origem do elo j-1
% escrito no sistema de coordenadas da base.
p0c1=r_01+R_01*rc11;
p0c2=r_02+R_01*R_12*rc22;
p0c3=r_03+R_01*R_12*R_23*rc33;
p0c4=r_04+R_01*R_12*R_23*R_34*rc44;
p0c5=r_05+R_01*R_12*R_23*R_34*R_45*rc55;
p0c6=r_06+R_01*R_12*R_23*R_34*R_45*R_56*rc66;

p1c1=[0;0;0];
p1c2=r_12+R_12*rc22;
p1c3=r_13+R_12*R_23*rc33;
p1c4=r_14+R_12*R_23*R_34*rc44;

```

```

p1c5=r_15+R_12*R_23*R_34*R_45*rc55;
p1c6=r_16+R_12*R_23*R_34*R_45*R_56*rc66;

p2c1=[0;0;0];
p2c2=[0;0;0];
p2c3=r_23+R_23*rc33;
p2c4=r_24+R_23*R_34*rc44;
p2c5=r_25+R_23*R_34*R_45*rc55;
p2c6=r_26+R_23*R_34*R_45*R_56*rc66;

p3c1=[0;0;0];
p3c2=[0;0;0];
p3c3=[0;0;0];
p3c4=r_34+R_34*rc44;
p3c5=r_35+R_34*R_45*rc55;
p3c6=r_36+R_34*R_45*R_56*rc66;

p4c1=[0;0;0];
p4c2=[0;0;0];
p4c3=[0;0;0];
p4c4=[0;0;0];
p4c5=r_45+R_45*rc55;
p4c6=r_46+R_45*R_56*rc66;

p5c1=[0;0;0];
p5c2=[0;0;0];
p5c3=[0;0;0];
p5c4=[0;0;0];
p5c5=[0;0;0];
p5c6=r_56+R_56*rc66;

% sub-matriz Jacobiana associada à velocidade linear do centro de
massa
Jv1=[cross(z_0,p0c1), [0;0;0] , [0;0;0] , [0;0;0]
, [0;0;0] , [0;0;0]];
Jv2=[cross(z_0,p0c2), cross(z_1,p1c2), [0;0;0] , [0;0;0]
, [0;0;0] , [0;0;0]];
Jv3=[cross(z_0,p0c3), cross(z_1,p1c3), cross(z_2,p2c3), [0;0;0]
, [0;0;0] , [0;0;0]];
Jv4=[cross(z_0,p0c4), cross(z_1,p1c4), cross(z_2,p2c4), cross(z_3,p3c
4), [0;0;0] , [0;0;0]];
Jv5=[cross(z_0,p0c5), cross(z_1,p1c5), cross(z_2,p2c5), cross(z_3,p3c
5), cross(z_4,p4c5), [0;0;0]];
Jv6=[cross(z_0,p0c6), cross(z_1,p1c6), cross(z_2,p2c6), cross(z_3,p3c
6), cross(z_4,p4c6), cross(z_5,p5c6)];

% sub-matriz Jacobiana associada à velocidade angular
Jw1=[z_0, [0;0;0], [0;0;0], [0;0;0], [0;0;0], [0;0;0]];
Jw2=[z_0, z_1 , [0;0;0], [0;0;0], [0;0;0], [0;0;0]];
Jw3=[z_0, z_1 , z_2 , [0;0;0], [0;0;0], [0;0;0]];
Jw4=[z_0, z_1 , z_2 , z_3 , [0;0;0], [0;0;0]];
Jw5=[z_0, z_1 , z_2 , z_3 , z_4 , [0;0;0]];
Jw6=[z_0, z_1 , z_2 , z_3 , z_4 , z_5];

% Rotação das matrizes de inércia dos elos para sistema fixo
Ig1=R_01*Ig11*R_01';
Ig2=R_02*Ig22*R_02';
Ig3=R_03*Ig33*R_03';
Ig4=R_04*Ig44*R_04';
Ig5=R_05*Ig55*R_05';

```

```

Ig6=R_06*Ig66*R_06';

% Matriz de Inércia do Manipulador
M=Jv1'*m1*Jv1+Jw1'*Ig1*Jw1 + Jv2'*m2*Jv2+Jw2'*Ig2*Jw2 +
Jv3'*m3*Jv3+Jw3'*Ig3*Jw3 + Jv4'*m4*Jv4+Jw4'*Ig4*Jw4 +
Jv5'*m5*Jv5+Jw5'*Ig5*Jw5 + Jv6'*m6*Jv6+Jw6'*Ig6*Jw6;

end

```

Função para cálculo da cinemática inversa

```

% Cinemática Inversa

clear;
clc;

% Parametros de Denavit-Hartenberg
alfa1=pi/2;
alfa2=0;
alfa3=pi/2;
alfa4=-pi/2;
alfa5=pi/2;
alfa6=0;

a1=0.1;
a2=0.25;
a3=0;
a4=0;
a5=0;
a6=0;

d1=0;
d2=0;
d3=0;
d4=0.16;
d5=0;
d6=0.05;

% Posição desejada da extremidade
% q=[+0.25;-0.32;+0.20] % Posição de Testes 1
% q=[+0.40;+0.00;-0.16] % Posição de Testes 2
% q=[+0.35;+0.30;-0.05] % Posição de Testes 3

% Orientação da Extremidade
u=[0; 0; 1]
v=[0;-1; 0]
w=[1; 0; 0]

% Cálculo da Cinemática inversa
qx=q(1);
qy=q(2);
qz=q(3);

wx=w(1);
wy=w(2);
wz=w(3);

```

```

ux=u(1);
uy=u(2);
uz=u(3);

vx=v(1);
vy=v(2);
vz=v(3);

p=[qx-d6*wx; qy-d6*wy; qz-d6*wz; 1];

px=p(1);
py=p(2);
pz=p(3);

Theta1=atan(py/px); % Solução de Theta 1

c1=cos(Theta1);
s1=sin(Theta1);

k1=px^2+py^2+pz^2-2*a1*px*c1-2*a1*py*s1+a1^2-a2^2-d4^2;
k2=2*a2*d4;
Theta3=asin(k1/k2); % Solução de Theta 3

c3=cos(Theta3);
s3=sin(Theta3);

mi1=a2+d4*s3;
mi2=-d4*c3;
gama1=px*c1+py*s1-a1;
gama2=pz;

s2=(mi1*gama2-mi2*gama1)/(mi1^2+mi2^2);
c2=(mi1*gama1+mi2*gama2)/(mi1^2+mi2^2);

Theta2=atan2(s2,c2); % Solução de Theta 2
c2=cos(Theta2);
s2=sin(Theta2);

s23=sin(Theta2+Theta3);
c23=cos(Theta2+Theta3);

Theta5=acos(wx*c1*s23+s1*s23*wy-c23*wz); % Solução de Theta 5
c5=cos(Theta5);
s5=sin(Theta5);

c4=(wx*c1*c23+wy*s1*c23+wz*s23)/s5;
s4=(wx*s1-wy*c1)/s5;

Theta4=atan2(s4,c4); % Solução de Theta 4
c4=cos(Theta4);
s4=sin(Theta4);

c6=-(ux*c1*s23+uy*s1*s23-uz*c23)/s5;
s6=(vx*c1*s23+vy*s1*s23-vz*c23)/s5;

Theta6=atan2(s6,c6); % Solução de Theta 6
c6=cos(Theta6);
s6=sin(Theta6);

```



```

% Resultados dos Thetas em graus
Theta_g=[Theta1*180/pi;Theta2*180/pi;Theta3*180/pi;Theta4*180/pi;T
heta5*180/pi;Theta6*180/pi]

% Cinemática Direta para verificação
A_01=[[cos(Theta1),-
sin(Theta1)*cos(alfa1),sin(Theta1)*sin(alfa1),a1*cos(Theta1)];[sin
(Theta1),cos(Theta1)*cos(alfa1),-
cos(Theta1)*sin(alfa1),a1*sin(Theta1)];[0,sin(alfa1),cos(alfa1),d1
];[0,0,0,1]];
A_12=[[cos(Theta2),-
sin(Theta2)*cos(alfa2),sin(Theta2)*sin(alfa2),a2*cos(Theta2)];[sin
(Theta2),cos(Theta2)*cos(alfa2),-
cos(Theta2)*sin(alfa2),a2*sin(Theta2)];[0,sin(alfa2),cos(alfa2),d2
];[0,0,0,1]];
A_23=[[cos(Theta3),-
sin(Theta3)*cos(alfa3),sin(Theta3)*sin(alfa3),a3*cos(Theta3)];[sin
(Theta3),cos(Theta3)*cos(alfa3),-
cos(Theta3)*sin(alfa3),a3*sin(Theta3)];[0,sin(alfa3),cos(alfa3),d3
];[0,0,0,1]];
A_34=[[cos(Theta4),-
sin(Theta4)*cos(alfa4),sin(Theta4)*sin(alfa4),a4*cos(Theta4)];[sin
(Theta4),cos(Theta4)*cos(alfa4),-
cos(Theta4)*sin(alfa4),a4*sin(Theta4)];[0,sin(alfa4),cos(alfa4),d4
];[0,0,0,1]];
A_45=[[cos(Theta5),-
sin(Theta5)*cos(alfa5),sin(Theta5)*sin(alfa5),a5*cos(Theta5)];[sin
(Theta5),cos(Theta5)*cos(alfa5),-
cos(Theta5)*sin(alfa5),a5*sin(Theta5)];[0,sin(alfa5),cos(alfa5),d5
];[0,0,0,1]];
A_56=[[cos(Theta6),-
sin(Theta6)*cos(alfa6),sin(Theta6)*sin(alfa6),a6*cos(Theta6)];[sin
(Theta6),cos(Theta6)*cos(alfa6),-
cos(Theta6)*sin(alfa6),a6*sin(Theta6)];[0,sin(alfa6),cos(alfa6),d6
];[0,0,0,1]];

A_06=A_01*A_12*A_23*A_34*A_45*A_56;

q_d=[A_06(1,4);A_06(2,4);A_06(3,4)] % Posição da Extremidade pela
direta

% Erro na extremidade entre sol. direta e inversa
erro=q_d-q

```

Apêndice C – Dados dos motores e redutores

Junta 1:

Redutor: GBPH-0902-NP-020-AA341-500	
Motor: AK85H/3.75-1.8	
Ângulo de Passo (°)	1,8
Passo por volta	200
Num. de fases	2
Tensão Nominal(V)	3,75
Corrente por fase (A)	5,0
Resistência por fase(Ω)	0,75
Indutância por fase (mH)	6,4
Holding Torque (kgf.cm)	52
Detent Torque (kgf.cm)	2,6
Inércia Total(kg.m ²)	$2,00 \times 10^{-2}$
Coef. Atrito Total(kg.m/s)	$5,00 \times 10^{-2}$
Massa Motor (kg)	2,3
Massa Redutor (kg)	4,2
Relação de Transmissão	1/20
Fluxo magnético máximo(Vs)	0,019

Junta 2:

Conjunto 2	
Redutor: GBPH-0902-NP-020-AA341-500	
Motor: AK85H8/3.36-1.8	
Ângulo de Passo (°)	1,8
Passo por volta	200
Num. de fases	2
Tensão Nominal(V)	3,36
Corrente por fase (A)	4,2
Resistência por fase(Ω)	0,8
Indutância por fase (mH)	3,5
Holding Torque (kgf.cm)	42
Detent Torque (kgf.cm)	2,1
Inércia Total(kg.m ²)	$2,00 \times 10^{-2}$
Coef. Atrito Total(kg.m/s)	$5,00 \times 10^{-2}$
Massa Motor (kg)	2,3
Massa Redutor (kg)	4,2
Relação de Transmissão	1/20
Fluxo magnético máximo(Vs)	0,014

Junta 3

Motor com Redutor: AK57H/3G20-1.8	
Ângulo de Passo (°)	1,8
Passo por volta	200
Num. de fases	2
Tensão Nominal(V)	2,4
Corrente por fase (A)	3,0
Resistência por fase(Ω)	1,6
Indutância por fase (mH)	4,0
Holding Torque (kgf.cm)	15
Detent Torque (kgf.cm)	0,6
Inércia Total(kg.m ²)	$1,57 \times 10^{-2}$
Coef. Atrito Total(kg.m/s)	$5,00 \times 10^{-2}$
Massa Total (kg)	1,3
Relação de Transmissão	1/20
Fluxo magnético máximo(Vs)	0,01

Junta 4:

Motor com Redutor: AK57H/3G10-1.8	
Ângulo de Passo (°)	1,8
Passo por volta	200
Num. de fases	2
Tensão Nominal(V)	2,4
Corrente por fase (A)	3,0
Resistência por fase(Ω)	1,6
Indutância por fase (mH)	4
Holding Torque (kgf.cm)	15
Detent Torque (kgf.cm)	0,25
Inércia Total(kg.m ²)	$1,57 \times 10^{-2}$
Coef. Atrito Total(kg.m/s)	$5,00 \times 10^{-2}$
Massa Total (kg)	1,3
Relação de Transmissão	1/10
Fluxo magnético máximo(Vs)	0,010

Junta 5:

Motor: AK56H/3-1.8	
Redutor: 2 pares de Polias	
Ângulo de Passo (°)	1,8
Passo por volta	200
Num. de fases	2
Tensão Nominal(V)	4,32
Corrente por fase (A)	2,4
Resistência por fase(Ω)	3,6
Indutância por fase (mH)	4,8
Holding Torque (kgf.cm)	7,6
Detent Torque (kgf.cm)	0,25
Inércia Rotor (kg.m ²)	$3,00 \times 10^{-5}$
Inércia Polias(kg.m ²)	$1,90 \times 10^{-5}$
Inércia Total(kg.m ²)	$1,25 \times 10^{-2}$
Coef. Atrito Total(kg.m/s)	$5,00 \times 10^{-2}$
Massa Motor (kg)	0,6
Relação de Transmissão	1/10
Fluxo magnético máximo(Vs)	0,010

Junta 6:

Motor: AK56H/3-1.8	
Redutor: 2 pares de Polias	
Ângulo de Passo (°)	1,8
Passo por volta	200
Num. de fases	2
Tensão Nominal(V)	4,32
Corrente por fase (A)	2,4
Resistência por fase(Ω)	3,6
Indutância por fase (mH)	4,8
Holding Torque (kgf.cm)	7,6
Detent Torque (kgf.cm)	0,4
Inércia Rotor (kg.m ²)	$3,00 \times 10^{-5}$
Inércia Polias(kg.m ²)	$1,90 \times 10^{-5}$
Inércia Total(kg.m ²)	$1,25 \times 10^{-2}$
Coef. Atrito Total(kg.m/s)	$5,00 \times 10^{-2}$
Massa Motor (kg)	0,6
Relação de Transmissão	1/10
Fluxo magnético máximo(Vs)	0,0094