

Universidade de Brasília – UnB
Faculdade de Tecnologia – FT
Departamento de Engenharia Elétrica – ENE
Relatório de Atividades de Bolsa de Iniciação Científica – CnPq
Período: Abril 2003 a março de 2004

PROJETO E IMPLEMENTAÇÃO DE UMA NOVA ARQUITETURA DE CONTROLE PARA O ROBÔ RHINO XR-4

Aluna: Carla Silva Rocha Aguiar
Curso de Graduação em Engenharia Mecatrônica – 8º Semestre
Orientadores: Prof. Victor Hugo Casanova Alcalde
Prof. Geovany Araújo Borges

Índice

Índice	2
1 Introdução	3
1.1 Resumo Bibliográfico	4
1.2 Esboço do trabalho.....	6
2 Motor Elétrico de Corrente Contínua	7
2.1 Modelamento matemático de motor de corrente contínua controlado por armadura ..	7
2.2 Representação matemática no domínio –s.....	9
2.2.1 Função de transferência	9
2.2.2 Diagrama de blocos	10
3 Acionamento das juntas do Robô Rhino XR-4	12
3.1 Características dos motores Pittman	12
3.2 Circuito de acionamento dos motores.....	13
3.2.1 Circuito de acionamento dos motores com controle de corrente	13
3.2.2 Modulador PWM.....	14
3.2.3 Ponte H	15
3.2.4 Etapa de testes.....	15
3.2.5 Circuito de acionamento dos motores e de Leitura das variáveis de controle..	16
4 Arquitetura de Controle do Robô Rhino XR- 4	21
4.1 Arquitetura de Controle para as Juntas do Robô Rhino XR-4 (PIC-SERVO)	21
4.2 Descrição da nova Arquitetura de Controle para as Juntas do Robô Rhino XR-4 ...	22
4.2.1 Especificações da placa de aquisição A-821PGL	24
4.2.2 Circuito de Controle	27
4.2.3 Geração de PWM por meio do Contador 82C54	28
4.2.4 Determinação da posição, velocidade e aceleração angular de cada junta a partir do controlador de Interrupção programável.....	30
4.2.5 Operação	31
5 Sistema de Controle	32
5.1 Arquitetura de Controle.....	32
5.1.1 Plataforma Rtx no Windows NT.....	32
5.2 Desenvolvimento do algoritmo de controle	34
5.2.1 Resultados obtidos com a implementação de controle PID e Controle Adaptativo	35
5.2.2 Controle PID:	35
5.2.3 Controle Adaptativo das juntas.....	36
6 Conclusões	42

1 Introdução

Este trabalho apresenta o projeto de uma nova arquitetura de controle em tempo real para o manipulador Rhino XR-4. Esse sistema apresenta como principais vantagens um baixo custo, simplicidade da lógica operacional e uma melhor performance quando comparada ao sistema anteriormente usado baseados no controle via micro-controladores.

O sistema de controle em tempo real é indicado em situações no qual a rapidez de resposta do manipulador e a precisão de sua movimentação é requerida. Tais características são necessárias na maioria das aplicações industriais.

O projeto de uma nova arquitetura de controle foi baseado em diversos módulos. O primeiro consistiu na concepção de um novo sistema de acionamento de motores e leitura dos sensores. Em seguida foi projetado um sistema de transmissão de dados entre o computador e os sistema físico, no caso do robô Rhino seis motores de corrente contínua. A principal característica do sistema implementado é o fato de todo o controle do manipulador ser realizado pelo computador, em tempo real. Isso simplificou bastante a implementação de todo o hardware e permite ao usuário uma maior acessibilidade ao sistema.

Todo o controle das juntas foi realizado em ambiente *Windows NT* com a plataforma tempo real *RTX*. Foi implementado controle PID, o mesmo usado no antigo sistema com micro-controladores, e a comparação entre as duas arquiteturas foi feita para validar o sistema proposto. Para a melhoria da performance do manipulador, foi implementado um controle adaptativo nas juntas e seus resultados foram comparados com os obtidos com o controle PID.

1.1 Resumo Bibliográfico

Schilling, R. J. – Fundamentals of Robotics – Pretice Hall, 1990.

Foi o primeiro contato com o projeto. No primeiro capítulo, é discutido terminologias ligadas à robóticas como classificação e especificações de robôs. Como exemplo neste capítulo, é feita a classificação do robô educacional Rhino XR-3, uma versão mais antiga do robô Rhino XR-4.

O capítulo 2 é dedicado à cinemática direta e o capítulo 3 à cinemática inversa. Em ambos há exemplos de estudos destas cinemáticas aplicadas ao robô Rhino XR-3.

Ogata, Katsuhiko. – Engenharia de Controle Moderno – 3ª edição, Pretice Hall do Brasil Ltda, 1982.

No capítulo 4, há uma extensa discussão sobre o modelamento do motor de corrente contínua controlado por armadura. Faz ainda especificações de regime transitório e permanente de sistemas de primeira e segunda ordem.

No capítulo 5, são discutidos ações de controle básico como controladores PID, Proporcional, Integral e Derivativo, e seus efeitos sobre o desempenho de um sistema. Varias aplicações deste tipo de controlador é exemplificados.

Franklin, G. F, Powell, J.D., Workman, M.– Digital control for dynamic systems – 3ª edição, Addison Wesley Longman, 1998.

No capítulo 2, há uma breve revisão sobre controle contínuo no tempo, leis de controle, estabilidade, entre outras definições.

No capítulo 3 é introduzido o controle digital. Tem-se ênfase o processo de digitalização de sistemas contínuos e a definição de controle PID no controle discreto.

Astrom, K. J., Wittenmark, B., Workman.– Adaptive Control – 2ª edição, Addison Wesley Longman, 1934.

Este livro contém as premissas básicas do controle adaptativo. No capítulo 2, são expostos métodos para a estimação on-line de parâmetros e metodologia da realização dos experimentos para aquisição de dados para a estimação.

No capítulo 5 é apresentado o sistema adaptativo *modelo-referência*, que foi projetado e implementado como controlador de velocidade para as juntas do Rhino XR-4, baseado na teoria apresentada neste capítulo.

Matsumoto, E. Y. – Matlab 6.5 Fundamentos de Programação – 1ª edição, Editora Érica Ltda, 2002.

Este livro expõe diversos comandos e diversas aplicações do software Matlab. Foi utilizado durante todo o projeto como um guia de pesquisa. Vale ressaltar o capítulo 5, no qual é tratado a maipulação de gráficos e o capítulo 6, direcionado à manipulação da ferramenta GUIDE – *Graphic User Interface Design Environment*, para a criação de interfaces gráficas em ambiente Matlab

1.2 Esboço do trabalho

Este trabalho apresenta os resultados obtidos no projeto e implementação de uma nova Arquitetura de Controle para fazer o controle das juntas do Robô Manipulador Rhino XR-4.

O capítulo 2 é apresentada a dinâmica de um motor de corrente contínua controlado por armadura, as relações físicas entre as diversas variáveis e por fim é apresentado a função de transferência que relaciona a posição angular atingida pelo motor devida à aplicação de uma dada tensão de entrada. O mesmo sistema é também representado em forma de diagrama de blocos.

No capítulo 3 são introduzidas as características dos motores que movimentam as juntas do robô Rhino XR-4, motores Pittman GM9413K046 e GM8712E465. É apresentado também o projeto de dois circuitos de acionamento de motores DC, a vantagem e desvantagem de cada circuito e a escolha do que mais se adequa ao projeto.

A descrição da nova Arquitetura de Controle utilizada para o controle das juntas do robô Rhino XR-4 é feita no capítulo 4. São apresentados os componentes desta nova arquitetura e a sua respectiva função dentro desta. Os modos de comunicação entre os módulos são apresentados, assim como o fluxo dos dados na Arquitetura.

O Capítulo 5 é reservado a apresentar o controle implementado até o momento nas juntas do robô Rhino XR-4. São explicadas as bibliotecas desenvolvida para o controle das juntas e suas funções. Dados relativos ao controle de velocidade das juntas são obtidos e os resultados são analisados

As conclusões e perspectivas para o trabalho são expressas no capítulo 6.

2 Motor Elétrico de Corrente Contínua

O capítulo 2 consiste no modelamento do motor de corrente contínua controlado por armadura, representado pela sua função de transferência e pelo diagrama de blocos correspondente.

2.1 Modelamento matemático de motor de corrente contínua controlado por armadura

O motor de corrente contínua é um dispositivo que converte a energia elétrica, em forma de tensão e corrente, fornecida na entrada de seus terminais em energia mecânica, em forma de velocidade angular e torque.

Um motor de corrente contínua controlada por armadura (com a corrente de campo mantida constante e o controle do motor ser exercida pela corrente de armadura) pode ser representado pelo circuito elétrico e as relações mecânicas mostrado na figura 2.1.

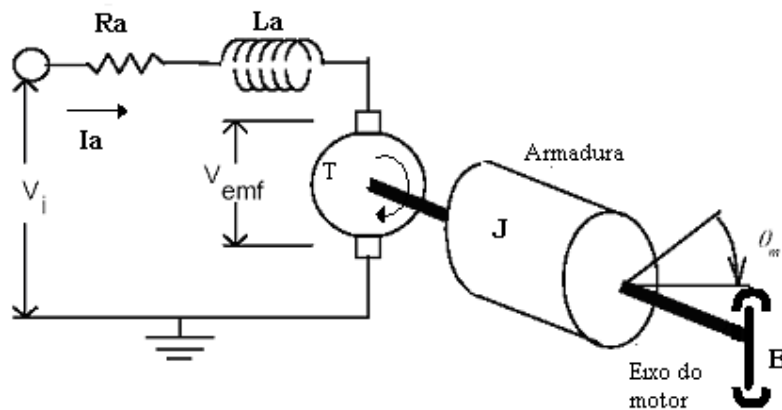


Figura 2.1 _ Esquemático de um motor de corrente contínua controlado por armadura

No qual

V_i – Tensão no terminal de entrada do motor DC [V];

R_a – Resistência de armadura [Ω];

L_a – Indutância de armadura [$\Omega.s$];

V_{emf} – força contra-eletromotriz [V];

I_a – corrente de armadura [A];

T – torque gerado pelo motor [N.m];

J – momento de inércia [$N.m.s^2$];

Θ_m – posição angular [rad];

B – coeficiente do atrito viscoso [N.m.s].

A resistência de armadura R_a é devido às perdas no enrolamento de armadura e L_a é devido à indutância de dispersão da armadura.

Aplicando a Lei das malhas no circuito elétrico equivalente mostrado na figura 2.1, tem-se que a equação diferencial associada à armadura do motor de corrente contínua é definida por

$$v_i(t) = L_a \frac{di_a(t)}{dt} + R_a * i_a(t) + V_{emf}(t) \quad (\text{Eq.2.1})$$

A equação diferencial mecânica associada ao sistema representado na figura 2.1 é definida pela equação 2.2. Note que foi desconsiderada a presença de algum torque perturbador.

$$T(t) = J * \frac{d^2\Theta(t)}{dt^2} + B * \frac{d\Theta(t)}{dt} \quad (\text{Eq. 2.2})$$

As equações que relacionam as grandezas elétricas e mecânicas são definidas pelas equações 2.3 por 2.5.

$$V_{emf} = K_E * \frac{d\Theta(t)}{dt} \quad (\text{Eq.2.3})$$

Como $\frac{d\Theta(t)}{dt} = \omega(t)$, no qual $\omega(t)$ é a velocidade angular, tem-se que

$$V_{emf} = K_E * \omega(t) \quad (\text{Eq. 2.4})$$

No qual K_E é a constante contra-eletromotriz, expressa em V/rad/s.

Ainda tem-se que

$$T(t) = K_T . i_a \quad (\text{Eq.2.5})$$

no qual K_T é a constante de torque do motor, expressa em N-m/A.

2.2 Representação matemática no domínio –s

2.2.1 Função de transferência

Considerando todas as condições iniciais nulas, pode-se representar estas mesmas equações no domínio de Laplace (s), tendo as seguintes igualdades:

$$(L_a . s + R_a) . I_a(s) + V_{emf} = V_i(s) \quad (\text{Eq. 2.6})$$

$$(J . s^2 + B . s) \Theta(s) = T(s) = K_T . I_a(s) \quad (\text{Eq. 2.7})$$

$$K_E . s \Theta(s) = V_{emf}(s) \quad (\text{Eq.2.8})$$

Pela equação 2.7, nota-se que o torque desenvolvido pelo motor, T, é diretamente proporcional a corrente de armadura que passa no motor, I_a . Logo, para um, sistema de controle no qual uma das variáveis a serem controladas seja o torque desenvolvido pelo motor, ao invés de inserir sensores no sistema para medir o valor do torque, basta apenas fazer o controle da corrente que passa pelo motor.

Considerando como sinal de entrada a tensão sobre o motor, $V_i(s)$, e como saída a posição angular, $\theta(s)$, isola-se a corrente de armadura da equação 2.6 e 2.7 e substituindo V_{emf} da equação 2.6 pela equação 2.8, chega-se a seguinte função de transferência:

$$\frac{\Theta(s)}{V_i(s)} = \frac{K_T}{s [L_a . J . s^2 + (L_a . B + R_a . J) s + R_a . B + K_T . K_E]} \quad (\text{Eq. 2.9})$$

A relação entre a velocidade angular, $\omega(s)$, e a tensão de entrada, $V_i(s)$, é dado por

$$\frac{\omega(s)}{V_i(s)} = \frac{s . \Theta(s)}{V_i(s)} = \frac{K_T}{L_a . J . s^2 + (L_a . B + R_a . J) s + R_a . B + K_T . K_E} \quad (\text{Eq.2.10})$$

2.2.2 Diagrama de blocos

O motor de corrente contínua controlado por armadura pode ser representado também por diagrama de bloco, a partir das equações no domínio de Laplace, inserindo ao sistema um torque externo $T_e(s)$:

$$(L_a \cdot s + R_a) \cdot I_a(s) + V_{emf} = V_i(s) \quad (\text{Eq. 2.11})$$

$$(J \cdot s + B) \cdot \omega(s) + T_e(s) = T(s) = K_T \cdot I_a(s) \quad (\text{Eq. 2.12})$$

$$K_E \cdot \omega(s) = V_{emf}(s) \quad (\text{Eq. 2.13})$$

A equação 2.11 pode ser representado pelo seguinte diagrama de blocos:

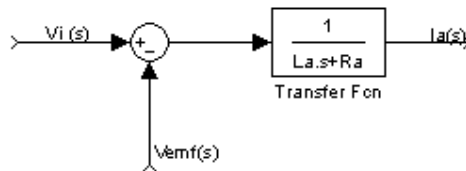


Figura 2.2

A equação 2.12 pode ser representado pelo seguinte diagrama de blocos:

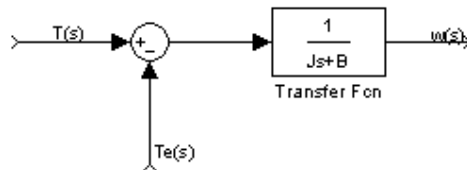


Figura 2.3

A partir dos diagramas de blocos mostrados nas figuras 2.2 e 2.3 e das relações expressas nas equações 2.11, 2.12, 2.13, obtém-se a representação por diagrama de blocos do motor DC controlado por armadura:

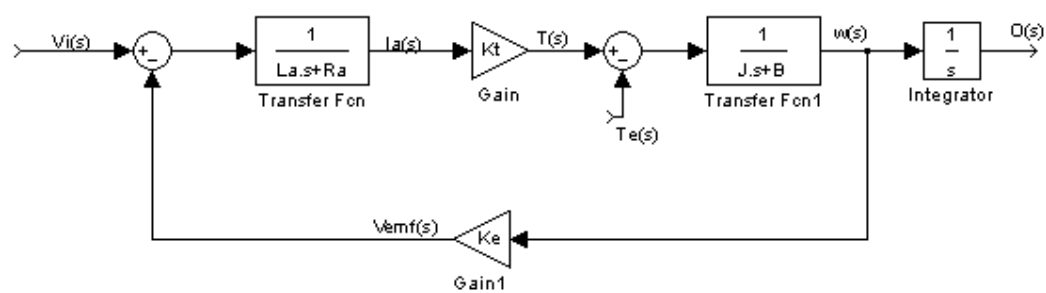


Figura 2.4

3 Acionamento das juntas do Robô Rhino XR-4

3.1 Características dos motores Pittman

A movimentação das juntas do robô educacional Rhino XR-4 se dá pelo acionamento dos motores da marca Pittman e modelos GM9413K046-R1, presentes nas juntas da base, do ombro, do cotovelo e do pulso, e GM8712E465, presentes na garra e no rolamento. De acordo com o fabricante, tais motores podem ser alimentados com tensões de até 24 V (GM8712E465) e 12 V (GM9413K046-R1) e a corrente de pico máxima sugerida para ambos motores é de 1,44 A.

Cada motor também possui um codificador óptico digital constituído de faixas opacas e dois sensores em quadratura, referenciados como canais A e B. A rotação do motor faz com que seja gerado sinais lógicos compatíveis com níveis TTL. Esses sinais podem ser interpretados por hardware e/ou software para se obter informações relacionado à junta tal como sentido, velocidade de rotação e a posição incremental.

O acionamento dos motores, logo das juntas do robô, se dá por meio de modulação PWM (*Pulse Width Modulation* – Modulação de Largura de Pulso), ou seja, a excitação do motor é uma onda retangular que assume valores discretos de zero e do valor de alimentação dos motores, no caso 12 V.

A resposta em frequência de um motor de corrente contínua é semelhante à um filtro passa baixa: se a frequência da forma de onda de excitação for maior que a frequência de corte do motor DC, este responderá como se uma tensão igual ao valor médio da forma de onda de excitação fosse aplicado à entrada, ou seja, para variar a tensão média sobre o motor, basta apenas variar o ciclo útil da onda de excitação PWM. Caso, porém, a frequência da forma de onda de excitação for compatível com a frequência de corte do motor DC, o motor tentará seguir a variação de tensão na entrada de seus terminais, apresentando um ruído sonoro para baixas velocidades e pequenas variações (*ripple*) sobre a velocidade média desenvolvida.

A alimentação dos terminais do motor e do circuito do codificador óptico, assim como os sinais do encoder (canais A e B) são disponibilizados nos motores Pittman por meio de um soquete do tipo IDC10, e a função referente à cada pino é mostrada na tabela 3.1.

Pino	Descrição
1	Saída do motor (M+)
2	Saída do motor (M+)
3	Saída do motor (M-)
4	Saída do motor (M-)
5	Saída do sensor de fim de curso
6	Terra do chassi
7	Canal B do encoder
8	Alimentação do circuito de encoder (+5 V)
9	Canal A do encoder
10	Terra para a fonte de alimentação do encoder

Tabela 3.1 _ Descrição dos pinos do motor do robô

3.2 Circuito de acionamento dos motores

No desenvolvimento do trabalho, foram projetados dois circuitos para o acionamento de motor de corrente contínua controlado por armadura. Primeiro desenvolveu-se um circuito genérico para o acionamento do motor com o controle da corrente implementado analogicamente. Em seguida, foi projetado um circuito de acionamento mais simples, sem controle de corrente implementado, tentando satisfazer as especificações dos motores Pittman.

3.2.1 Circuito de acionamento dos motores com controle de corrente

O acionamento dos motores, se dá por meio de modulação PWM. Para aplicar tal onda nos terminais do motor, decidiu-se projetar um circuito de acionamento que houvesse um gerador de sinal PWM, que variasse seu ciclo de trabalho de 0% à 100% em nível TTL (tensão discreta de 0 V e 5 V) de acordo com a tensão de referência _ que seria uma das entradas do sistema. Uma ponte H amplificaria o sinal PWM, agora assumindo tensão de 0 V e 12 V, de tal forma a fornecer corrente o suficiente para acionar o motor.

Foi estabelecido que a entrada do sistema seria uma corrente de referência para o dado motor e o sentido de rotação do mesmo. A corrente de referência é dada por uma tensão correspondente que varia de 0 V à 3 V e o sentido de rotação assume dois valores discretos: 0 V (sentido horário) e 5 V (sentido anti-horário).

O circuito projetado é mostrado na figura 3.1. Além das características já citadas, fizemos o controle em malha fechada da corrente que passa no motor. Após alguns testes, percebeu-se que um controlador Proporcional atendia a necessidade de a corrente no motor seguir a referência com o período transitório estável o bastante que não apresentasse picos de correntes (oscilações) que pudessem danificar ou o circuito ou o próprio motor. Implementou-se também um limitador de corrente que permitia o máximo de 2 A no motor.

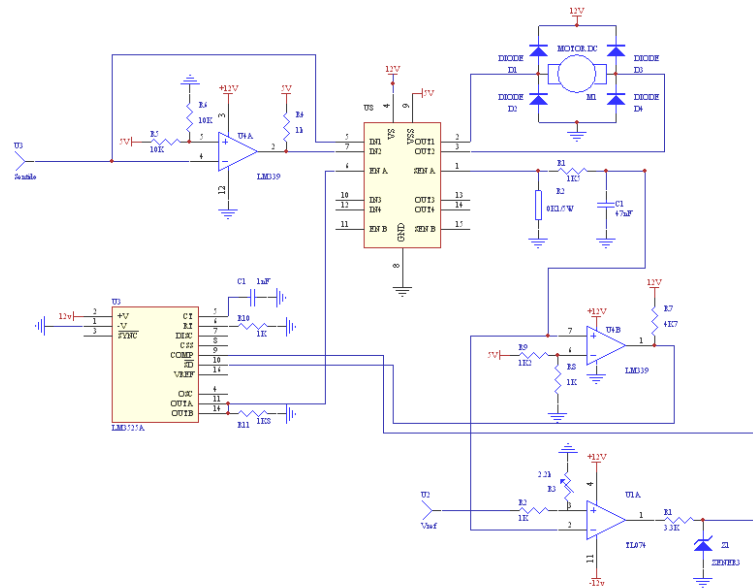


Fig 3.1 _ Circuito projetado para acionamento dos motores e controle proporcional da corrente em cada motor.

3.2.2 Modulador PWM

O circuito responsável pela geração do ciclo PWM é mostrado na figura 3.1. Optou-se pela escolha do circuito integrado de modulador PWM *LM3524D*, da *National Semiconductor*, uma vez que este permite configurar a frequência da forma da onda de saída PWM por meio da escolha da resistência e da capacitância conectadas aos pinos 6 e 7 respectivamente. Esta configuração é necessária, pois deve-se configurar a frequência do ciclo de trabalho PWM maior que a frequência de corte do motor DC.

A escolha dos valores do resistor e do capacitor para determinar a frequência do ciclo de trabalho PWM foi realizada empiricamente, testando para o motor tanto em aberto quanto com carga, utilizando como referência o gráfico fornecido pelo fabricante que relaciona a frequência de oscilação com os valores da resistência e da capacitância (Fig. 3.2). Para os motores Pittman, escolheu-se a resistência igual a 1 K Ω e a capacitância igual a 1nF, obtendo a frequência de oscilação de cerca de 550 KHz.

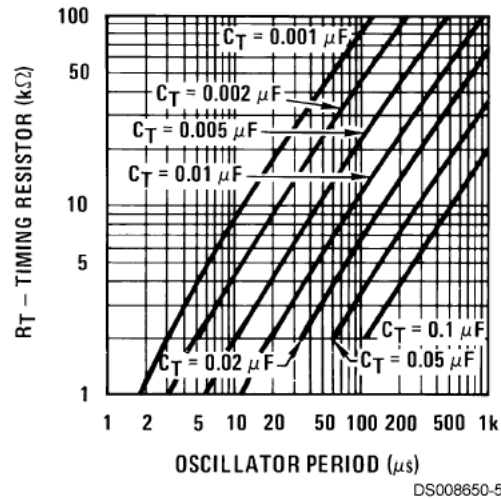


Figura 3.2 _ Relação entre a resistência , a capacitância e o período oscilatório do modulador PWM LM3524

3.2.3 Ponte H

A função de um circuito de ponte H é fornecer aos terminais do motor tensão e corrente o suficiente para acionar os motores. Para a escolha adequada do circuito a ser empregado para este fim, necessita saber das especificações dos motores a ser acionado. No caso dos motores Pittman GM9413K046-R1 e GM8712E465, estes são alimentados por uma tensão de 0V à 12V e 24 V, respectivamente, e tem o limite de corrente com carga de 1.44 A. Com base nestas especificações, foi escolhido o circuito integrado L298, que além de possuir duas pontes H em seu encapsulamento, possui um pino de sensor de corrente para cada ponte, o que possibilitou fazer o controle de corrente analogicamente (por meio de circuitos com amplificadores operacionais) . O circuito da configuração da Ponte H e do controle de corrente é mostrado da figura 3.1. Observa-se que a habilitação da Ponte H é dada pelo sinal de PWM e as entradas (*In*) determinam o sentido de rotação do motor.

O circuito foi inicialmente implementado em protoboard para fins de teste e posteriormente este foi confeccionado em circuito impresso.

3.2.4 Etapa de testes

A etapa seguinte foi testar tal circuito com os motores Pittman e determinar se a resposta destes satisfaziam as condições impostas: a corrente no motor deveria seguir a corrente de referência, o período transitório deveria ser baixo, a tensão média sobre os motores deveria ser igual ao valor médio da forma de onda de excitação fosse aplicada à entrada, e o circuito limitador de corrente deveria limitar a corrente em 2 A no motor.

Quanto ao controle de corrente, o resultado foi satisfatório. O controlador proporcional foi suficiente para fazer com que a corrente no motor seguisse a referência e que esta corrente não apresentasse sobressinal (o sobressinal pode ser percebido quando aplicou-se um controlador Proporcional e Integral _ PI). Porém, a calibração na comparação entre a corrente no motor e a referência é feita por pela variação da resistência do trimpote, o que não oferece um controle preciso, apesar dos resultados obtidos serem satisfatórios.

Pelo fato de a ponte H ser habilitada pelo sinal PWM, observou-se que a tensão média sobre os motores não era igual ao valor médio da forma de onda de excitação aplicada à entrada, pois quando a tensão do ciclo PWM era igual a 0V a ponte H era desabilitada, o que fazia com que os terminais do motor ficassem em aberto. Isso fazia com que a tensão sobre o motor fosse diminuindo, quando que o esperado era que a tensão sobre o motor fosse para 0V .

O circuito limitador de tensão funcionou como esperado, limitando a corrente no motor em 2 A e mantendo esta corrente até que a referência de corrente diminuísse.

Os resultados obtidos por meio de testes foram considerados satisfatórios a princípio. Porém, uma nova proposta de controle das juntas do robô Rhino XR-4 foi feita: implementar o controle das juntas no computador e em tempo real, deixando para o hardware apenas a tarefa de acionar os motores e fazer a leitura dos sinais das variáveis a serem controladas (corrente, sinais dos encoders, fim-de-curso) e condicioná-las de modo a poderem ser enviadas para o PC.

3.2.5 Circuito de acionamento dos motores e de Leitura das variáveis de controle

A primeira característica do novo circuito de acionamento é que este não fará qualquer tipo de controle sobre as variáveis de controle (corrente, velocidade, posição angular). Além disso, definiu-se o ciclo PWM seria gerado por meio da programação adequada do contador 82C54. Com isso, a entrada para este novo circuito seria o ciclo PWM para cada motor e o sentido de rotação de cada motor e como saída do sistema, seria disponibilizado as variáveis de controle.

O mesmo CI L298 foi escolhido como ponte H, mas o circuito teve que ser modificado para garantir que a tensão média sobre o terminal do motor fosse igual ao valor médio da forma de onda de excitação aplicada à entrada. O circuito implementado é mostrado na . A porta lógica *AND* (74LS08) é usada como buffer para a proteção do sistema e a porta lógica *XOR* (74LS86) é usada para determinar o sentido de rotação do motor.

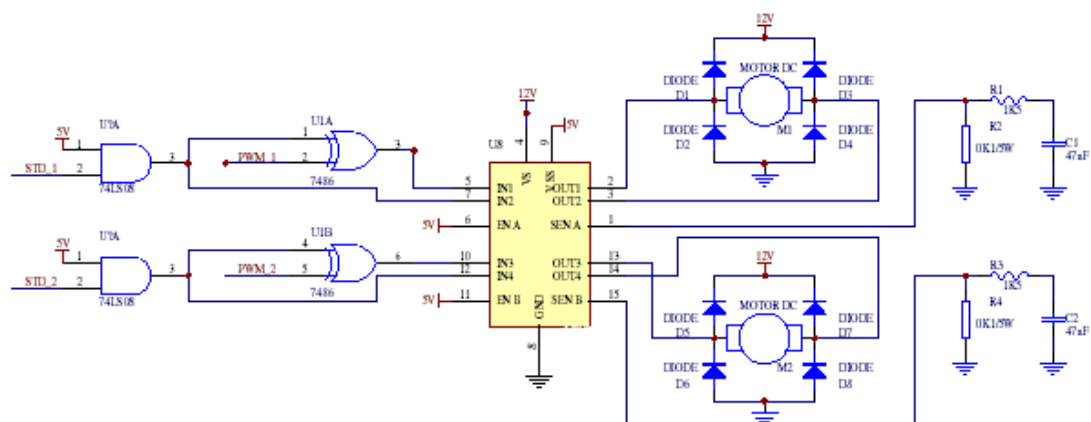


Figura 3.6 _ Circuito de acionamento de motor DC

Para fazer a interface de dados entre o circuito de acionamento e o circuito de interface (que será melhor explicado no Capítulo 4) foi inserido na placa do circuito de acionamento um conector de 40 pinos, cuja descrição é mostrada na tabela 3.2.

Pino	Descrição	Pino	Descrição
1	Sentido motor 1	2	PWM motor 1
3	Sentido motor 2	4	PWM motor 2
5	Sentido motor 3	6	PWM motor 3
7	Sentido motor 4	8	PWM motor 4
9	Sentido motor 5	10	PWM motor 5
11	Sentido motor 6	12	PWM motor 6
13	--	14	--
15	--	16	--
17	Sinal encoder 1 A	18	Sinal encoder 1B
19	Sinal sensor de corrente 1	20	Sinal de final de curso 1
21	Sinal encoder 2 A	22	Sinal encoder 2B
23	Sinal sensor de corrente 2	24	Sinal de final de curso 2
25	Sinal encoder 3A	26	Sinal encoder 3B
27	Sinal sensor de corrente 3	28	Sinal de final de curso 3
29	Sinal encoder 4 A	30	Sinal encoder 4B
31	Sinal sensor de corrente 4	32	Sinal de final de curso 4
33	Sinal encoder 5 A	34	Sinal encoder 5B
35	Sinal sensor de corrente 5	36	Sinal de final de curso 5
37	Sinal encoder 6 A	38	Sinal encoder 6B
39	Sinal sensor de corrente 6	40	Sinal de final de curso 6

Tabela 3.2 _ Descrição do conector da placa de acionamento

A implementação do circuito de acionamento dos seis motores do robô educacional Rhino XR-4 é mostrado na figura 3.7.

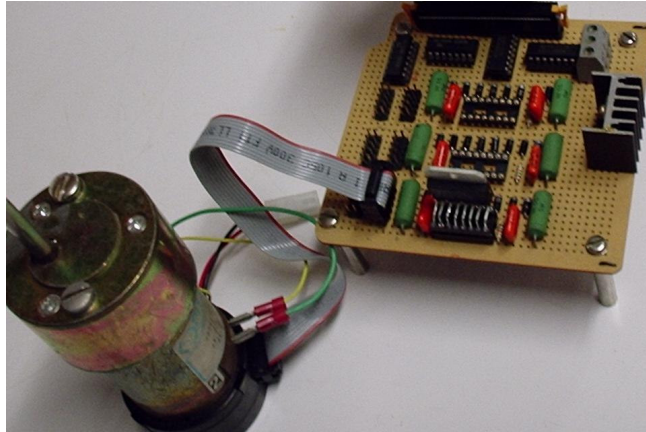


Figura 3.7 _ Foto do circuito de acionamento de seis motores

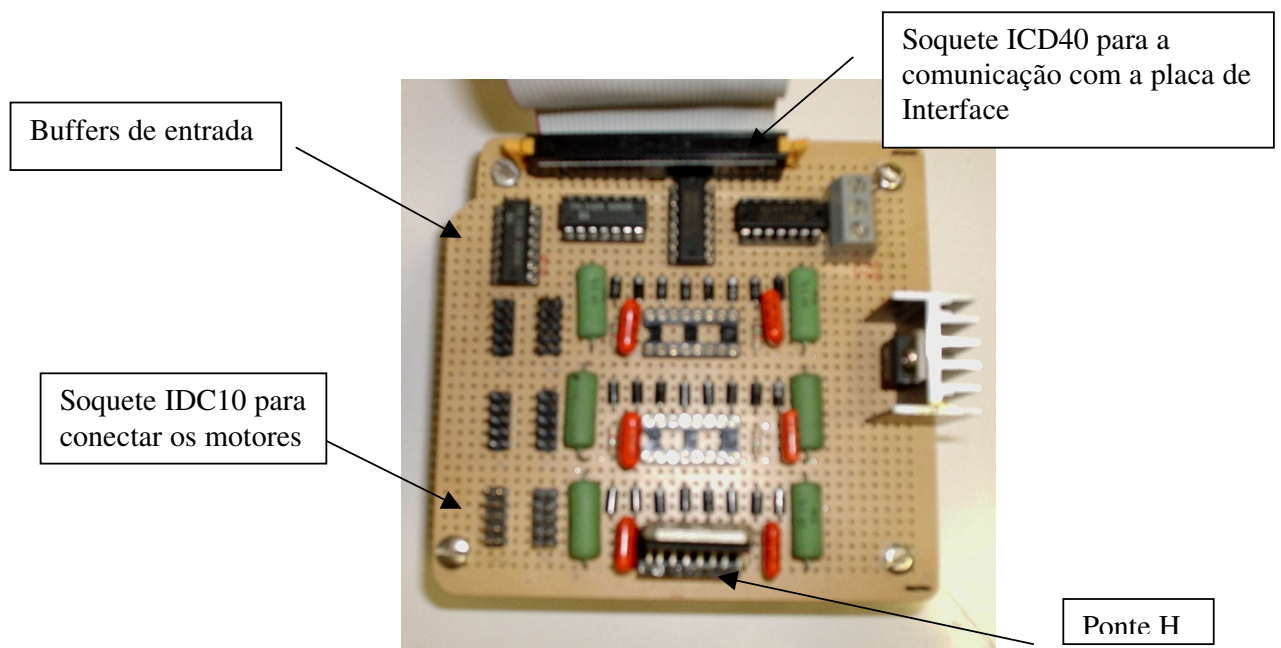


Figura 3.8 _ Foto detalhada do circuito de acionamento de motores

Testes realizados mostraram que a nova configuração do circuito fez com que a tensão média sobre o terminal do motor fosse igual ao valor médio da forma de onda de excitação aplicada à entrada. Além disso, os sinais digitais de saída do circuito, dos encoders e dos sensores de fim de curso, são compatíveis com a lógica TTL (onda quadrada assumindo valores discretos de 0 V e 5 V), graças à circuitos de condicionamento presentes nos codificadores ópticos digitais, o que possibilita a leitura direta pela placa de interface. Já os sinais analógicos de saída, dos sensores de corrente de cada motor, são tensões que variam de 0 volts ,para corrente de 0 A no motor, à 2 volts, para corrente de 2 A no motor, apresentou pouco ruído devido à presença de capacitores na saída destas tensões, condicionando o sinal

para ser lido pelo conversor analógico/digital presente na placa de aquisição (abordada no capítulo 4) sem que haja distorção da informação (valor da corrente do motor).

4 Arquitetura de Controle do Robô Rhino XR- 4

4.1 Arquitetura de Controle para as Juntas do Robô Rhino XR-4 (PIC-SERVO)

A primeira Arquitetura de Controle desenvolvida para o controle das seis juntas do robô educacional Rhino XR-4 foi baseada nas placas de controle PIC-SERVO, adquiridas em paralelo à aquisição da estrutura do robô, em 1997.

A placa de controle PIC-SERVO é um sistema de controle completo proporcionando o controle de motores DC com codificador óptico digital, incluindo controle PID e geração de trajetórias com perfis *trapezoidal* e de velocidade. Possui circuitos periféricos para a contagem dos pulsos gerados pelos encoders e amplificadores para fornecer correntes de até 3 A contínuos ao motor.

Nesta arquitetura, há uma placa de controle PIC-SERVO para cada junta, e todas estas placas são interligadas e realizavam a comunicação com o computador via porta serial RS232. Os dados referentes à uma dada trajetória, assim como os parâmetros PID (Proporcional Integral e Derivativo) relativos à cada junta, são enviadas em pacotes por meio da porta serial RS232 para o PIC-SERVO e todo o controle de trajetória é realizado por meio do microcontrolador PIC16C73. Este calcula o ciclo PWM, com a frequência do sinal fixa em 10 kHz, a partir do algoritmo de controle e este sinal é amplificado pelo LMD18201 para que o motor possa ser acionado.

Sinais dos codificadores óticos são tratados pelo microcontrolador, que é programado como um contador, PIC-ENC. Este fornece a realimentação de posição e de velocidade ao PIC16C73. Com estes dados, o PIC-SERVO recalcula o valor do ciclo PWM e o sentido de rotação de cada junta a partir do algoritmo de controle.

A comunicação serial no sistema de controle PIC-SERVO utiliza o padrão *full-duplex* (comunicação bidirecional simultânea) com um *start bit* (nível lógico 0), seguido por um byte de dados (bit menos significativo primeiro) e um *stop bit* (nível lógico 1). O envio de um byte de dados pela porta serial é realizado em cerca de dois milissegundos. Por esta razão, todas as informações relativas à uma dada trajetória e os valores dos ganhos do controlador PID devem ser enviadas para o PIC-SERVO antes que a trajetória seja iniciada.

4.2 Descrição da nova Arquitetura de Controle para as Juntas do Robô Rhino XR-4

Apesar da arquitetura de controle baseada no PIC-SERVO apresentar resultados satisfatórios na realização de trajetórias previamente determinadas pelo usuário, foi proposto o projeto de uma nova Arquitetura de Controle cujo enfoque principal fosse fazer desse novo Sistema de Controle o mais próximo dos implementados em robôs manipuladores comerciais.

Para o controle das juntas de manipuladores comerciais, principalmente os robôs com movimentos de precisão, são implementados não somente controladores do tipo PID, mas também controles mais robustos que consideram a inércia, o torque externo sobre o eixo do motor, o ruído na tensão de alimentação ou na leitura dos sensores, ou até mesmo a má determinação dos parâmetros estimados para o controle retroativo das juntas, fazendo com que a movimentação do robô seja de forma contínua. Tais tipos de controladores, como o controle adaptativo, por exemplo, são impossíveis de serem implementados na Arquitetura de Controle baseada no PIC-SERVO pela inflexibilidade de programação no microcontrolador.

Foi decidido então que todo o controle das juntas seriam realizadas pelo PC visto sua superioridade de processamento, a facilidade de programação de algoritmos de controle e o maior controle sobre as variáveis pelo o usuário. Os sinais para o controle de movimento rotacional, posição, velocidade, aceleração e torque, além de informações sobre o fim de curso do Rhino XR-4, devem ser lidas e interpretadas periodicamente para que o algoritmo de controle possa atuar na correção da trajetória. Por isso a comunicação entre o processo e o PC deve ser rápida e ocorrer periodicamente. Visando preencher este requisito, optou-se pela programação do código de controle sobre a plataforma em tempo real RTX no sistema operacional Windows NT, garantindo assim a periodicidade na aquisição e envio dos dados, e realizar a comunicação com o processo utilização da placa de aquisição A-821PGL, garantindo a rapidez no envio dos dados. Como exemplo comparativo: o envio de 8 bits de dados na Arquitetura de Controle baseada no PIC-SERVO utilizando a porta serial RS232 se dá em 2 milissegundos, o envio de 8 bits utilizando a porta paralela, programando sobre a plataforma em tempo real RTX, se dá em 32 microssegundos. E o envio de 16 bits de dados na nova Arquitetura de Controle se dá em 6 microssegundos.

O fato de todo o controle de trajetória do Rhino XR-4 ser centralizado no PC simplifica o Hardware envolvido para a aquisição das variáveis referentes à uma dada junta (a corrente, os sinais do encoder, o sinal de fim-de-curso), uma vez que tais dados serão apenas condicionados para serem lidos pelo PC, para posteriormente serem interpretados.

A nova Arquitetura de controle é composta dos seguintes componentes:

- Computador PC, utilizando o sistema operacional Windows NT com plataforma em tempo real Rtx;
- Placa de aquisição A-821PGL;
- Circuito de Interface;
- Circuito de acionamento de motor DC;
- Fonte de Potência para os circuitos lógicos e para acionar os motores (12 V);
- 6 cabos do tipo flat ribbon de 10 pinos, com soquetes IDC10 fêmea em ambos os lados para conectar os motores das juntas do Rhino XR-4 no circuito de acionamento;
- 1 cabo do tipo flat ribbon de 40 pinos, com soquetes IDC40 fêmea em ambos os lados para fazer a conexão entre o circuito de acionamento de motores e o circuito de Controle ;
- 2 cabos do tipo flat ribbon de 20 pinos, com soquetes IDC20 fêmea em ambos os lados para conectar o circuito de interface às entrada e saída digital da placa de aquisição A-821PGL;

As conexões entre os componentes da nova arquitetura de controle são mostradas na figura 4.1.

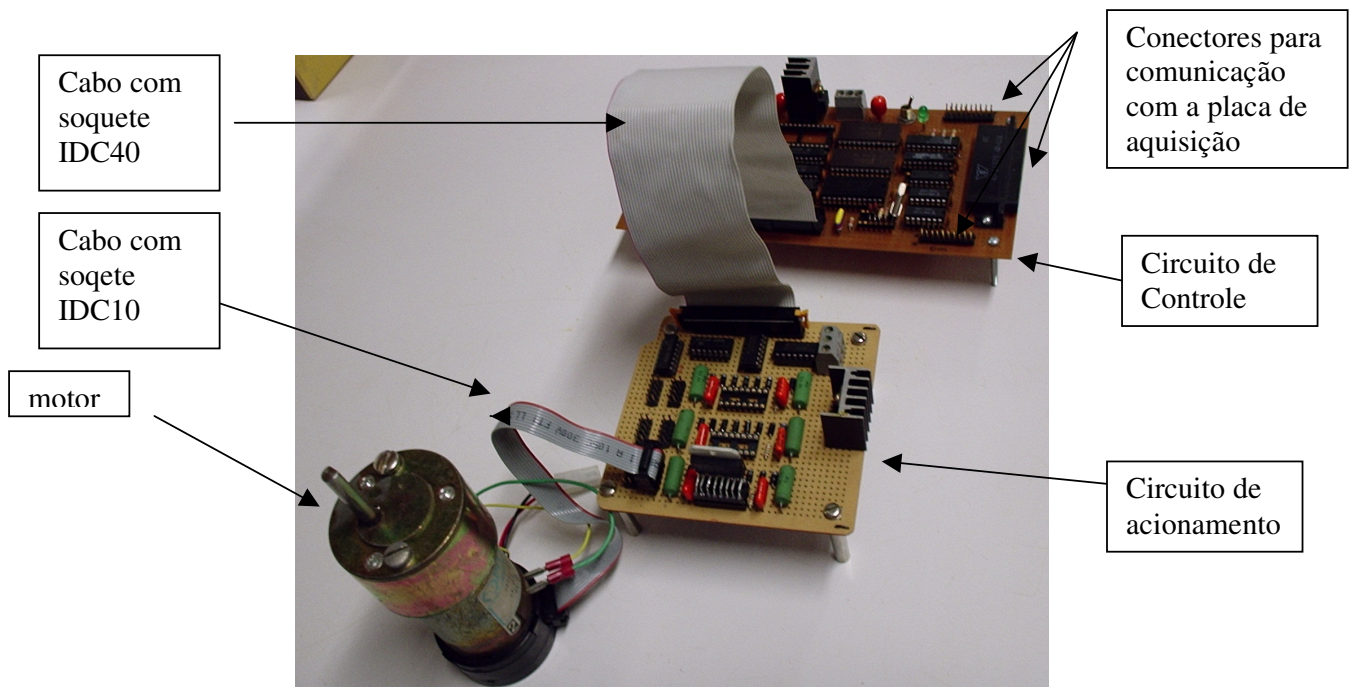


Figura 4.1 _ Ligações entre os componentes da Arquitetura de controle

4.2.1 Especificações da placa de aquisição A-821PGL

A A-821PGH é uma placa de aquisição, envio e tratamento de dados genérica fabricado pela ICP_{DAS} – Industrial Computer Product _ Data Acquisition System, com barramento ISA, que possui as seguintes características principais:

- Barramento ISA compatível com AT PC
- 8 sinais de entrada analógica diferencial ou 16 *single-ended*
- Ganho baixo Programável: 1,2,4,5
Entrada: $\pm 5V$, $\pm 2.5V$, $\pm 1.25V$, $\pm 0.625V$
- 1 cana de 12 bit D/A de saída de voltagem
- 16 entradas digitais e 16 saídas digitas (TTL)

A configuração física da placa de aquisição A-821PGL é mostrada na figura 4.2

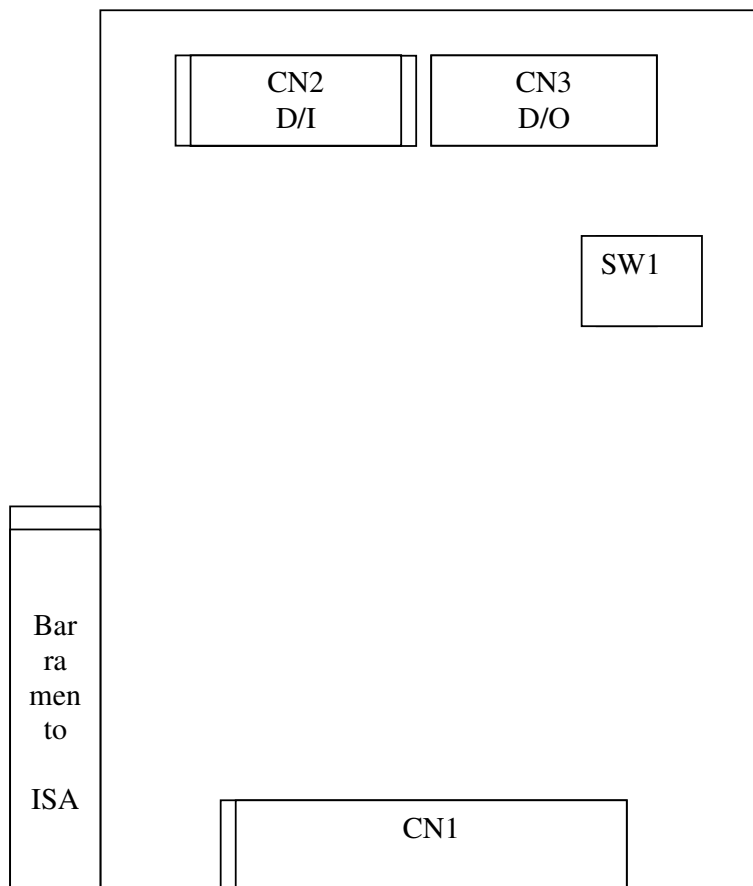


Figura 4.2_ Diagrama da placa de aquisição A-821PGL

Da figura 4.2 pode-se identificar:

CN1 – Conector de 37 pinos de entrada/saída analógica;

CN2 – Conector de 20 pinos de entrada digital;

CN3 – Conector de 20 pinos de saída digital;

SW1 – chaves de configuração do endereço de base de entrada/ saída.

A placa de aquisição A-821PGL ocupa 16 locais consecutivos no espaço de endereçamento de I/O. O endereço de base é definido pela combinação das posições das chaves do SW1, podendo assumir valor de 200-20F à 3FO-3FF. Para garantir a exclusividade do endereço de base da placa, escolhemos este como sendo 240-24F uma vez que nenhum dispositivo no computador no qual está sendo desenvolvido o projeto possui o endereço de base correspondente.

A pinagem dos conectores CN1, CN2 e CN3 serão mostrados nas tabelas 4.1, 4.2, 4.3, respectivamente.

Pino	Descrição	Pino	Descrição
1	Entrada Analógica 0/+	20	Entrada Analógica 8/+
2	Entrada Analógica 1/+	21	Entrada Analógica 9/+
3	Entrada Analógica 2/+	22	Entrada Analógica 10/+
4	Entrada Analógica 3/+	23	Entrada Analógica 11/+
5	Entrada Analógica 4/+	24	Entrada Analógica 12/+
6	Entrada Analógica 5/+	25	Entrada Analógica 13/+
7	Entrada Analógica 6/+	26	Entrada Analógica 14/+
8	Entrada Analógica 7/+	27	Entrada Analógica 15/+
9	GND analógico	28	GND analógico
10	GND analógico	29	GND analógico
11	Não usado	30	Canal D/A
12	Não usado	31	Não usado
13	Saída de 12 volts	32	Não usado
14	GND analógico	33	Não usado
15	GND analógico	34	Não usado
16	Não usado	35	Não usado

17	Não usado	36	Não usado
18	Não usado	37	Não usado
19	Saída de 5 volts		

Tabela 4.1 _ Descrição dos pinos do conector CN1

Pino	Descrição	Pino	Descrição
1	Entrada digital 0/TTL	2	Entrada digital 1/TTL
3	Entrada digital 2/TTL	4	Entrada digital 3/TTL
5	Entrada digital 4/TTL	6	Entrada digital 5/TTL
7	Entrada digital 6/TTL	8	Entrada digital 7/TTL
9	Entrada digital 8/TTL	10	Entrada digital 9/TTL
11	Entrada digital 10/TTL	12	Entrada digital 11/TTL
13	Entrada digital 12/TTL	14	Entrada digital 13/TTL
15	Entrada digital 14/TTL	16	Entrada digital 15/TTL
17	Saída do GND	18	Saída do GND
19	Saída de 5 volts	20	STROBE

Tabela 4.2 _ Descrição dos pinos do conector CN2

Pino	Descrição	Pino	Descrição
1	Saída digital 0/TTL	2	Saída digital 1/TTL
3	Saída digital 2/TTL	4	Saída digital 3/TTL
5	Saída digital 4/TTL	6	Saída digital 5/TTL
7	Saída digital 6/TTL	8	Saída digital 7/TTL
9	Saída digital 8/TTL	10	Saída digital 9/TTL
11	Saída digital 10/TTL	12	Saída digital 11/TTL
13	Saída digital 12/TTL	14	Saída digital 13/TTL
15	Saída digital 14/TTL	16	Saída digital 15/TTL
17	Saída do GND	18	Saída do GND
19	Saída de 5 volts	20	Saída de 12 volts

Tabela 4.3 _ Descrição dos pinos do conector CN3

Na nova Arquitetura de Controle é utilizada as 16 entradas de sinais digitais, as 16 saídas de sinais digitais (envio e recebimento de dados) e 6 entradas analógicas (leitura da

corrente de cada motor). Será utilizado posteriormente o conversor A/D (Analógico/ Digital) para fazer o controle da corrente sobre cada motor.

4.2.2 Circuito de Controle

A finalidade do circuito de Controle é gerar e enviar o sinal PWM e o sentido de rotação para o motor a partir do sinal de controle enviado do computador. Para que o computador possa fazer a leitura das variáveis, é enviado o sinal de controle indicando que qual a variável a ser lida posteriormente (como corrente, fim-de-curso, sentido de rotação), e por meio do circuito de Interface, este conteúdo é disponibilizado no barramento de saída.

O diagrama do circuito de Controle projetado e implementado é mostrado na figura 4.3.

O circuito de Controle foi projetado de modo a simular um barramento de 8 Bits. Isso faz com que o programa para o envio e leitura de dados seja feito de forma modular, sendo que é enviado 8 bits de controle e 8 bits de dados. Esta modularidade facilita a inserção de novos componentes ao circuito, uma vez que o envio de dados ocorre sempre da mesma maneira, o que muda é conteúdo do sinal de controle.

A implementação do circuito de Controle é mostrada na figura 4.4 e em seguida são explicados detalhadamente os principais componentes e suas funções no circuito.

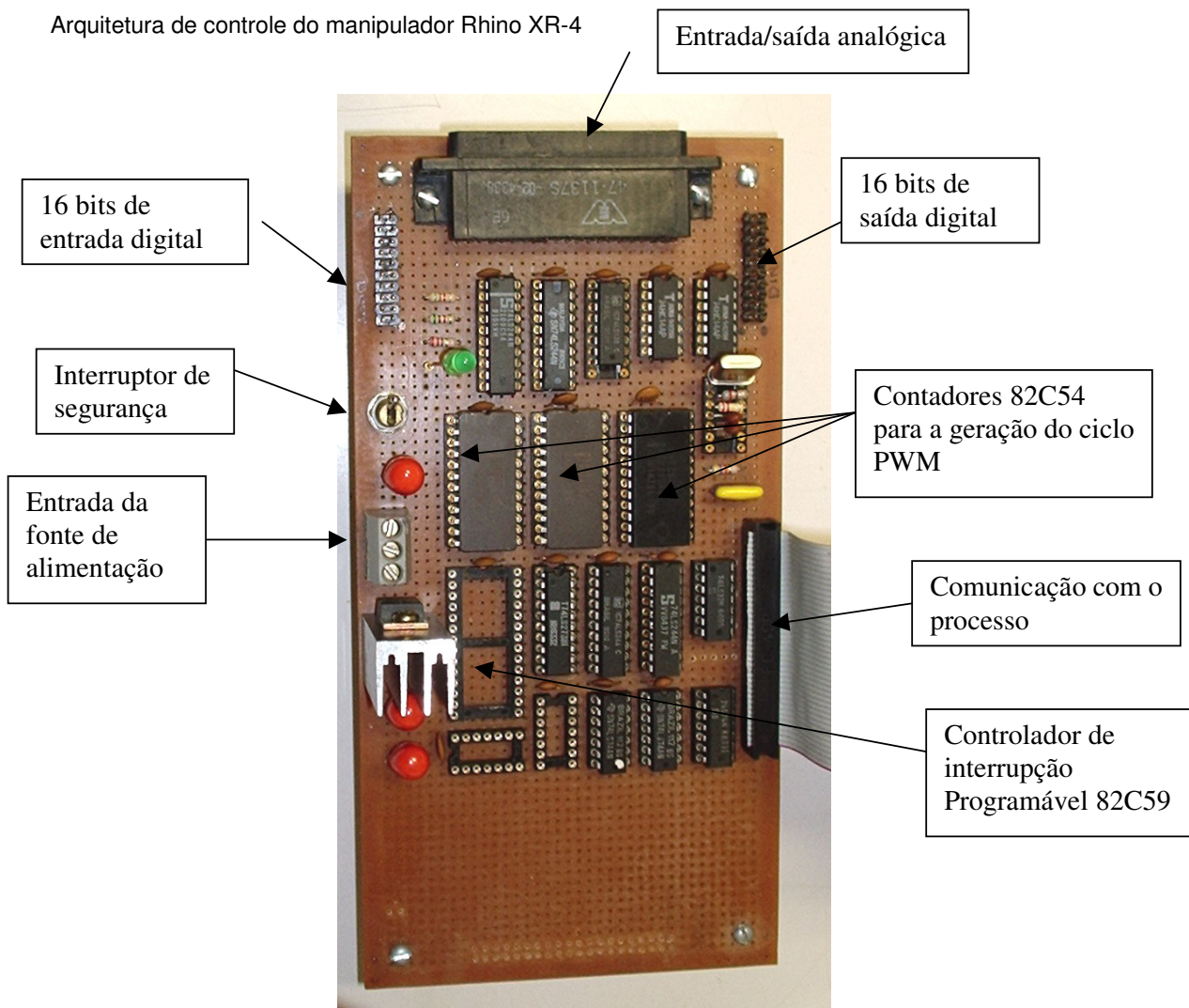


Figura 4.3 _ Foto do Circuito de Controle

4.2.3 Geração de PWM por meio do Contador 82C54

O 82C54 é um contador programável fabricado pela Intel. Em seu encapsulamento há três contadores independentes, que podem ser programados em quatro modos de operação distintos. Para cada contador, há uma porta de Gate, que dependendo do modo de operação habilita o contador, uma porta de clock e uma saída para do contador.

Para realizar alguma operação dos contadores, há a necessidade de enviar 5 bits de controle, indicando que a operação é de escrita e especifica se trata de uma operação de configuração do modo de operação de um dos contadores ou apenas a mudança do valor de contagem de um dos contadores, ou de leitura. Há também 8 bits de dados, que varia de função de acordo com o tipo de operação: para a configuração do modo de operação, os 8 bits de dados indica o modo de operação e o contador a ser programado, além da quantidade e a maneira como bits com a contagem serão enviados; para a mudança de valor da contagem, os

8 bits é o próprio valor da contagem e segue o padrão definido na configuração; para a leitura de dados, os dados esperados são acessíveis por meios dos 8 bits de dados.

Para gerar o ciclo PWM por meio do contador 82C54, foram necessários 3 CIs do 82C54, uma vez que é necessário um contador para cada motor. Todos os clocks dos contadores foram ligados a um mesmo oscilador de 10 MHz. Um dos contadores foi programado no modo 2 de operação e a saída desse contador foi ligado à entrada Gate dos demais contadores. No modo 2 de operação a saída do contador permanece em nível lógico alto do início do valor de contagem (determinado na configuração) até que este valor seja decrementado ao valor 1. Neste instante a saída vai para o nível lógico baixo por um ciclo de clock e então é iniciado a contagem e a saída volta ao nível lógico alto. A frequência do clock é de 10 MHz e o valor de contagem programado para geração o ciclo PWM dos motores Pittman é de 1024, o que gera como saída do contador uma onda de frequência de 10 KHz. O diagrama mostrando o estado das portas do contador no modo 2 é mostrado na figura 4.4.

Os contadores restantes são programados no modo de operação 1 e a partir do valor de contagem, gera o ciclo PWM correspondente. No modo de operação 1, a saída do contador é inicialmente nível alto, e vai para nível baixo após o “disparo” na entrada Gate e mantém o nível até a contagem atingir o valor 1, voltando assim para o nível lógico alto. O diagrama mostrando o estado das portas do contador no modo 1 é mostrado na figura 4.5. Com isso é gerado o ciclo PWM que varia de 100% , com o valor de contagem igual a 1 e 0%, com o valor de contagem igual a 1024. O sinal então é invertido e enviado para o circuito de acionamento dos motores.

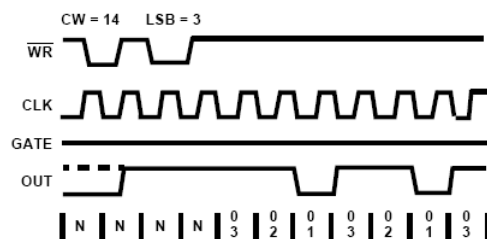


Figura 4.4 _ Modo 2 de operação

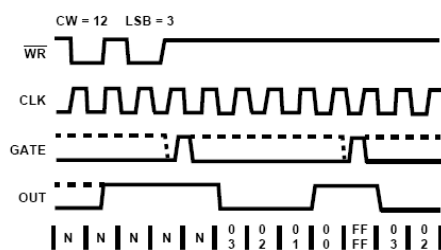


Figura 4.5 _ Modo 1 de operação

4.2.4 Determinação da posição, velocidade e aceleração angular de cada junta a partir do controlador de Interrupção programável.

O 82C59A é um controlador de interrupções programável que pode gerenciar até 8 interrupções simultâneas com diferentes prioridades. Seu princípio de funcionamento baseia-se no seguinte princípio: o microprocessador só interrompe a execução de uma tarefa da sua função principal quando uma interrupção é detectada e avaliada pelo 82C59 que indica além da ocorrência da interrupção, a origem da mesma, para que uma rotina de tratamento da interrupção específica seja realizada.

Quando ocorre uma interrupção, é setado a porta no 82C59 qual a interrupção ocorreu. Em seguida, é avaliada a prioridade da mesma e, se nenhuma outra interrupção estiver sendo tratada, é enviado um sinal de ocorrência de interrupção (pela porta INT) para a porta IRQ7 da porta paralela do computador. A confirmação da ocorrência de interrupção é feita pelo envio de um bit específico para o 82C59, setando a porta INTA com uma serie de pulsos. Com isso é disponibilizado no barramento de dados a fonte da interrupção, sendo lida, interpretada e gerado uma rotina de tratamento de interrupção pelo programa de controle.

A programação do controlador de interrupções 82C59 consiste na determinação de parâmetros como o vetor de endereçamento (para identificar a fonte da interrupção), a determinação do tipo de interrupção que vai ser considerada (por nível ou por subida), o tipo de prioridade entre as interrupções (prioridades iguais, prioridades definidas, etc), a rotina de tratamento de interrupções (quantos pulsos INTA serão enviados para o reconhecimento da interrupção), entre outros.

No caso específico da determinação do circuito de controle, nas entradas das interrupções era conectado o resultado da operação XOR (ou-exclusivo) dos sinais dos canais do encoder de cada motor. Na configuração do controlador de interrupção foi determinado que todas as interrupções teriam a mesma prioridade.

O tratamento da interrupção consiste em, a partir do conhecimento da origem da interrupção (motor que gerou o pulso no encoder), determinar o período entre as duas últimas interrupções, para na função de controle poder ser calculado a velocidade e a aceleração, e a partir da quantidade de interrupções ocorridas, considerando o sentido de rotação do motor em cada momento, determinar a posição angular da junta a cada instante.

4.2.5 Operação

A operação do sistema de controle do motor DC utilizando a nova Arquitetura de Controle pode ser considerada basicamente como envio de sinais de controle por parte do algoritmo de controle. O envio de dados por meio da porta de saída digital da placa de aquisição A-821PGL se dá por meio do envio de dois blocos de dados: 8 bits de controle são enviados pelas oito menos significativos bits da porta de saída digital, e 8 bits de dados são enviados pelos 8 bits mais significativos.

Os oito bits de controle indicam qual o dispositivo a ser operado (que é determinado pelo multiplexador 74LS138 ou pelo bit de entrada A0) e qual o tipo de operação será realizada (específico a cada dispositivo). Para operação de escrita, os oito bits de dados vão para o barramento de dados do circuito de controle e seu conteúdo depende do dispositivo e do tipo de operação a ser realizada. Para operações de leitura, o dado a ser lido fica disponível no barramento de dados até que este seja lido pelo algoritmo de controle.

Os sinais vindos dos motores são condicionados no circuito de acionamento e posteriormente no circuito de controle por meio de buffers (que podem ser portas lógicas ou simplesmente capacitores), e ficam disponíveis para a leitura (que ocorre periodicamente) no barramento de dados de saída. No o caso da interrupção, estes eventos ocorrem assincronamente, então seus dados só ficam disponíveis no barramento de dados quando uma interrupção válida é ocorrida e vai ser tratada.

5 Sistema de Controle

5.1 Arquitetura de Controle

Tem-se na figura 5.1 a arquitetura de controle do robô Rhino. Nesta arquitetura, liga-se diferentes sensores e acionadores ao computador central do robô. Este robô é conectado à esses diferentes sensores por meio da placa de aquisição **A-821PGL** ligado ao barramento ISA.

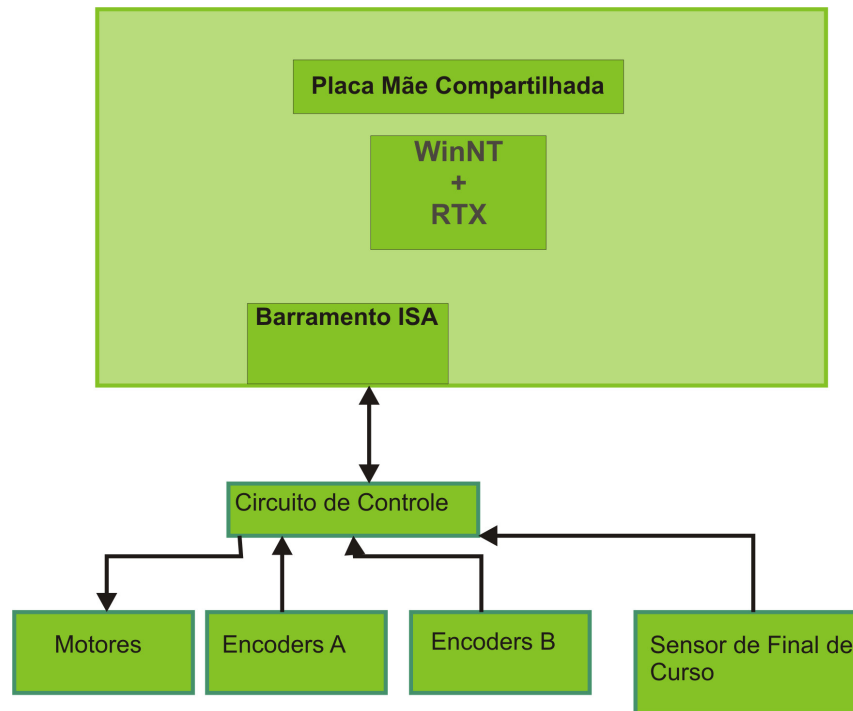


Figura 5.1 _ Organização Arquitetura de controle do robô Omni .

5.1.1 Plataforma Rtx no Windows NT

Windows NT (ou *WinNT*) é um sistema operacional multi-tarefas/ multi-usuários que utiliza um organizador round-robin preemptivo com prioridade dinâmica. Este organizador tenta maximizar a densidade das tarefas ativas. *RTX* (*Real Time eXtension*) , como o próprio nome diz, é uma extensão tempo real do *Windows NT*, que se integra neste último de acordo com o ilustrado na figura 5.2. O *RTX* é um sub-sistema que tem o seu próprio noyau, um organizador preemptivo com prioridades fixas e estrutura de controle e sincronização de tarefas.

As características principais do sistema *WindNT/RTX* são os seguintes:

- O noyau do *WindNT* é considerado como tarefa de fundo do *RTX*. Ele é executado sempre que não tem uma tarefa *RTX* ativa, por maior que seja a sua prioridade. Como

a interface gráfica é completamente controlada pelo *WinNT*, as funções de impressão de caracteres tais como **printf** são não determinísticas, uma vez que sua execução é gerado pelo *WinNT*. Por isso, o *RTX* tem a biblioteca **RTAPI_W32**, que inclui algumas funções do **API** (*Application Interface*) do *WinNT* que são reescritas de forma a se tornarem determinísticas.

- Uma extensão da HAL (*Hardware Abstraction Layer*) permite acesso direto ao material. Sem o *RTX*, acessar os dispositivos de entrada/saída são muitos mais complexos.
- As operações em ponto flutuante são permitidas dentro das tarefas *RTX*, o que não é o caso de dispositivos sobre o noyau do *WinNT*.

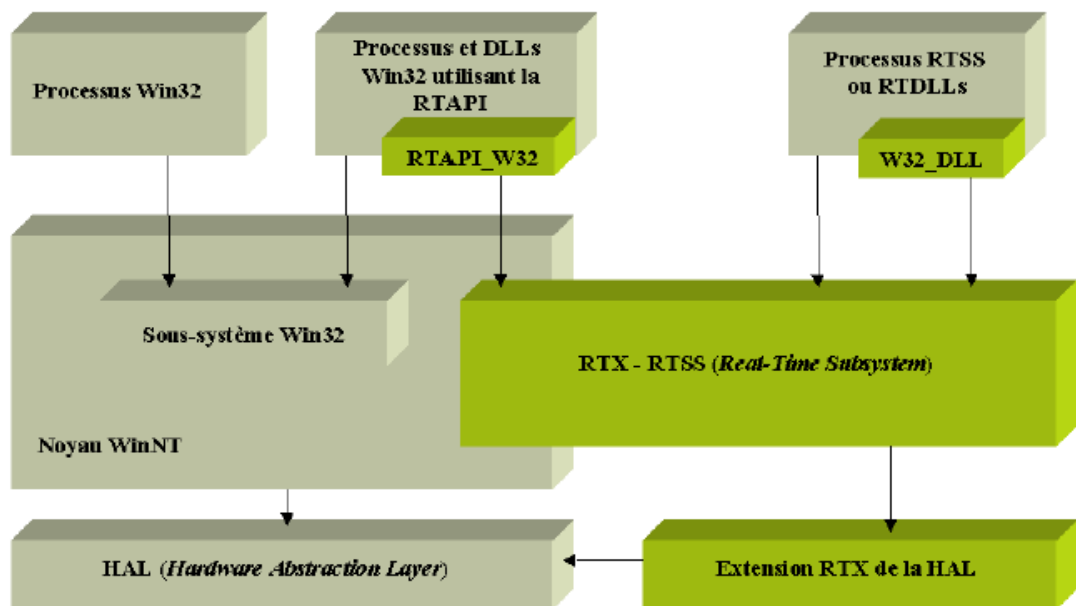


Figura 5.2 _ Organização dos módulos que compõem o conjunto *WinNT-RTX* .

As funcionalidades integradas no RTX são: os objetos de comunicação interprocessos (por exemplo semáforos, mutex, memória compartilhadas, etc) , os temporizadores, os relógios e as interrupções e o acesso à arquivos. É ressaltado que estes são as ferramentas mínimas necessária para a concepção de um sistema tempo real. Se necessário, novas ferramentas tais como caixas de mensagens devem ser criadas pelo usuário. RTX não contém suporte visual ; todo o acesso deve ser feito pelo WinNT. Isso limita bastante a utilização de tal sistema em aplicações comandados à distância e que precisam de um tempo de resposta curtos e determinísticos.

5.2 Desenvolvimento do algoritmo de controle

Após o projeto e a implementação da Arquitetura de Controle, iniciou uma nova fase de desenvolvimento do algoritmo de controle das juntas do robô manipulador Rhino XR-4. Este algoritmo está sendo programado no Microsoft Visual C++ sobre a plataforma tempo real, RTX.

A primeira biblioteca a ser desenvolvida foi a “RhinoInterf.c”. Nela estão todas as funções de mais baixo nível, relacionado à comunicação com o circuito de controle via placa de aquisição A-821PGL. Exemplos de funções contidas nesta biblioteca estão a configuração e inicialização dos contadores 82C54 e do controlador de interrupções 82C59 e funções de envio de ciclo de trabalho PWM e leitura do conteúdo dos dispositivos.

Testes foram realizados para determinar a taxa de transmissão de dados (16 bits são enviados em 6 μ s), a integridade dos sinais nos dispositivos (alto ruído foi detectado no circuito por causa fonte de alimentação sendo necessário inserir capacitores na fonte de alimentação e uma malha de terra) e a corretude das ligações feitas nos circuitos.

A segunda Biblioteca a ser desenvolvida, e ainda em fase de teste, é a “Rhino.c”. Nela há funções de mais alto nível e já direcionadas ao controle das juntas do Rhino XR-4. Exemplos de funções contidas nesta biblioteca:

- Tarefa em tempo real *TaskRhinoJointControl*: esta tarefa é realizada a cada 1 ms e realiza as seguintes operações. Faz-se a leitura do final de curso de todos os motores e para o motor da junta que estiver no final de curso, calcula a velocidade e a posição angular de todos os motores , aplica um filtro de segunda ordem na velocidade de cada junta e caso o controle esteja no modo velocidade, aplica o controle adaptativo em todas as juntas (o controle dos outros modos de controle ainda não foram implementados _ controle de corrente e posição).
- Funções para a leitura das diversas variáveis relativas à uma junta específica, como, por exemplo, todos os dados relativos a uma junta (velocidade, posição, fim de curso) ou a um motor (velocidade, corrente, tensão normalizada).
- Funções para leitura de parâmetros específicos ao Rhino e que estão armazenados em um arquivo texto, como a redução das engrenagens ou os Ticks por revolução de cada junta . É lido também os coeficientes do filtro de segunda ordem que será aplicado à velocidade de cada junta e os ganhos do controlador PID.

No programa principal, “RhinoRTX.c”, são programadas funções para testes das bibliotecas e para a aquisição e armazenamento dos dados referentes as juntas para análise off-line.

A tarefa em tempo real *TimerHandler* realizada a cada 5 ms faz a aquisição dos dados de todas as juntas e ao final da realização de um dado experimentos todas estes dados são gravados como matrizes em um arquivo *.mat determinado pelo o usuário. A biblioteca que manipula os dados, as armazena como matrizes e grava em um arquivo *.mat é a “*GMatrix.h*”, desenvolvida pelo professor Geovany Borges.

Os resultados obtidos pela implementação do controlador PID e do controlador adaptativo são mostrado no item 5.2.1 do capítulo 5, assim como a explicação dos controladores implementados.

Após implementar o controle de posição e de corrente, iniciará a elaboração da terceira biblioteca de controle do Rhino XR-4, no qual será a implementação do controle de trajetória para o manipulador.

5.2.1 Resultados obtidos com a implementação de controle PID e Controle Adaptativo

Foram implementados até o momento o controle PID (Proporcional, integral, derivativo) e o controle adaptativo das as juntas para o modo de controle de velocidade. Os resultados encontrados para o controle PID são mostrados no que se segue.

5.2.2 Controle PID:

Foi implementado um controle PID simples no qual a função de controle é dada por:

$$\frac{Y(s)}{E(s)} = \left(Kp + Kd * s + \frac{Ki}{s} \right) \quad (5.1)$$

$$Y(s) = \left(Kp * E(s) + Kd * E(s) * s + Ki * \frac{E(s)}{s} \right)$$

Que no domínio do tempo torna-se:

$$y(t) = Kp * e(t) + Kd * \frac{de(t)}{dt} + Ki * \int_0^t e(\tau) d\tau \quad (5.2)$$

Discretizando, obtém-se:

$$y(k) = Kp * e(k) + Kd * \left(\frac{e(k) - e(k-1)}{\Delta T} \right) + Ki * ei \quad (5.3)$$

no qual

$$e(k) = V_{ref} - V_{medido} \quad (5.4)$$

$$ei = ei + e(k) * \Delta T \quad (5.5)$$

5.2.3 Controle Adaptativo das juntas

O modelo-referência em sistema adaptativo (MRSA) é um tipo importante de controlador adaptativo. Ele pode ser visto como um sistema adaptativo de servo no qual a performance desejada é expressa em termos de modelo de referência, que fornece a resposta desejada devido à um sinal de controle. O diagrama de blocos desse sistema é mostrado na figura 5.3. O sistema tem uma realimentação simples composta do processo e do controlador e de uma outra realimentação que varia o parâmetro por meio do controlador de parâmetros. Os parâmetros são variados baseados no sinal do erro, diferença entre a saída do sistema e a saída do modelo de referência. A realimentação do sistema é chamada de *inner loop* e a realimentação do ajuste de parâmetro é chamada de *outer loop*.

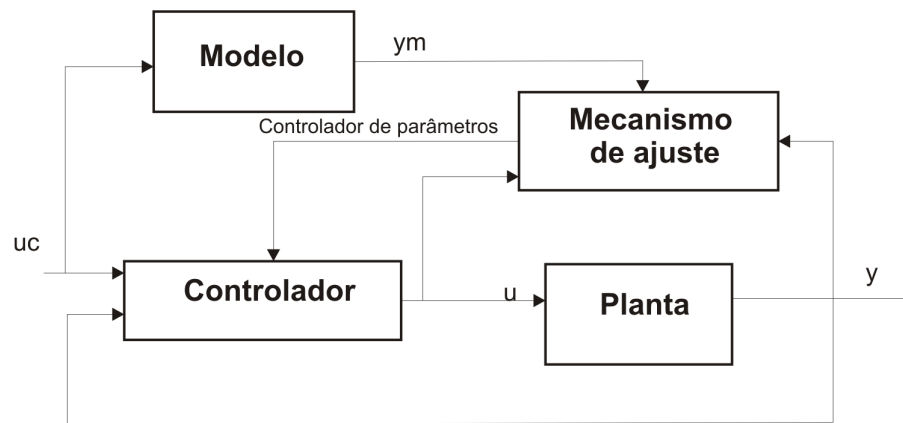


Figura 5.3 _ Diagrama de blocos de um modelo-referência do sistema adaptativo (MRSA) .

O mecanismo de ajuste dos parâmetros é obtido da seguinte maneira. Considerando que a resposta obtida pelo motor pode ser aproximada por um sistema de primeira ordem descrita por :

$$\begin{aligned} \frac{dy}{dt} &= -ay + bu \\ y(s) &= \frac{b}{s+a} u(s) \end{aligned} \quad (5.6)$$

No qual u é o sinal de referência de entrada e y é a medida da saída, no caso a velocidade sobre o motor. Assumindo que a resposta desejada para saída do sistema é dado por :

$$\frac{dy_m}{dt} = -a_m y_m + b_m u_c$$

E o controlador dado por :

$$u(t) = \theta_1 u_c(t) - \theta_2 y(t) \quad (5.7)$$

Tem-se que o controlador tem dois parâmetros. Se for escolhido

$$\begin{aligned} \theta_1 &= \theta_1^0 = \frac{b_m}{b} \\ \theta_2 &= \theta_2^0 = \frac{a_m - a}{b} \end{aligned} \quad (5.8)$$

A relação entrada/saída do sistema e do modelo são os mesmos. Isso é chamado de perfeito seguidor do modelo.

Introduzindo o erro ao sistema

$$e = y - y_m$$

No qual y é a saída do sistema com realimentação. Da equação 5.6 e 5.7 tem-se

$$y = \frac{b\theta_1}{p + a + b\theta_2} u_c$$

No qual $p = d/dt$ é o operador diferencial. A sensibilidade é obtida pelas derivadas parciais com respeito aos parâmetros do controlador θ_1 e θ_2 :

$$\begin{aligned} \frac{\partial e}{\partial \theta_1} &= \frac{b}{p + a + b\theta_2} u_c \\ \frac{\partial e}{\partial \theta_2} &= \frac{b^2 \theta_1}{(p + a + b\theta_2)^2} u_c = -\frac{b}{p + a + b\theta_2} y \end{aligned}$$

Os parâmetros de controle, θ_1 e θ_2 podem ser atualizados a partir de :

$$\begin{aligned} \frac{d\theta_1}{dt} &= -\gamma \left(\frac{a_m}{p + a_m} u_c \right) e \\ \frac{d\theta_2}{dt} &= \gamma \left(\frac{a_m}{p + a_m} y \right) e \end{aligned} \quad (5.9)$$

Nessas equações combinou-se os parâmetros b e a_m com o ganho adaptativo γ' , que aparece como o produto $\gamma' = \frac{\gamma a_m}{b}$. O sinal do parâmetro b deve ser conhecido para ter-se o

sinal correto do ganho adaptativo γ . Nota-se que o filtro foi normalizado de modo que em regime o ganho seja unitário.

O controlador adaptativo é um sistema dinâmico com cinco variáveis de estados que podem ser escolhidos para ser o modelo de saída, pelos parâmetros e a sensibilidade. O diagrama de blocos completa do controlador adaptativo é mostrado na figura 5.4.

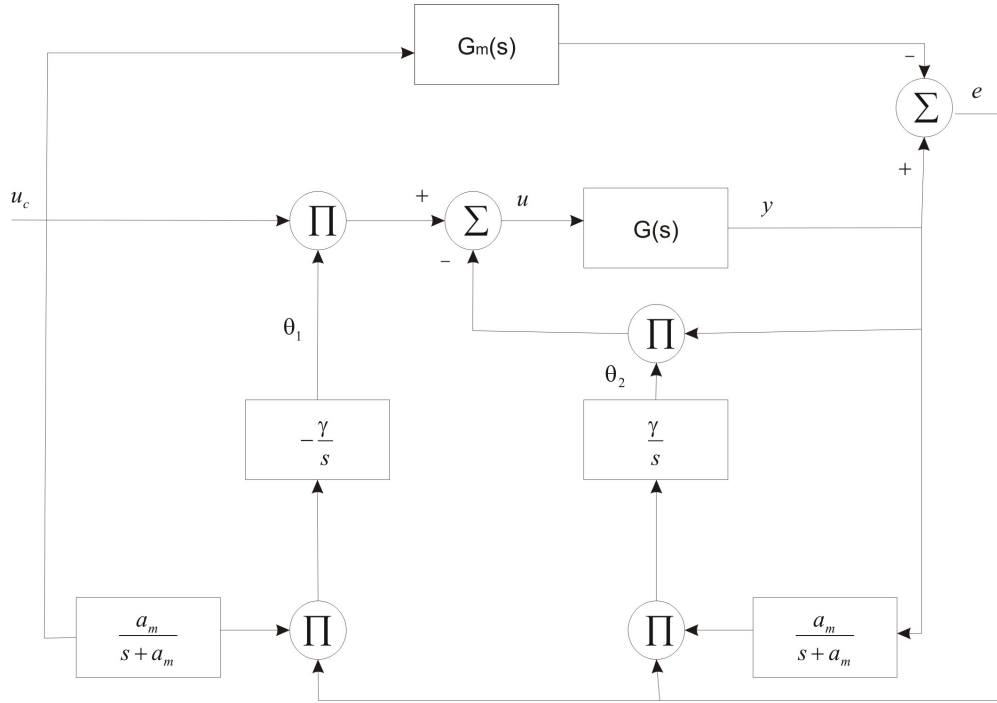


Figura 5.4 _ Diagrama de blocos de um controlador modelo-referência para um processo de primeira ordem.

5.2.3.1 Projeto do controlador adaptativo para as juntas do Rhino

A fim de implementar o controle adaptativo nas juntas do robô Rhino, fez-se necessário a determinação on-line dos parâmetros referentes à cada motor, que são representados pelos parâmetros b e a_m na equação 5.6, além dos valores iniciais dos parâmetros adaptativos que são γ , θ_1 e θ_2 . O ponto chave na identificação de sistemas é a seleção do modelo que melhor representa o processo (no caso, o acionamento dos motores), a elaboração de um experimento de identificação, a estimação paramétrica e a validação dos resultados da estimação.

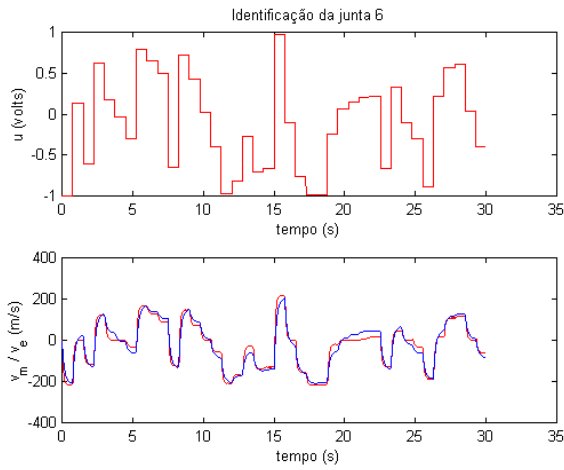
Considerando que a identificação do sistema é executada automaticamente em sistemas adaptativos, é essencial ter um bom entendimento do problema, e no caso estudado, o modelamento matemático do motor de corrente contínua é mostrado no capítulo 2.

Como mostrado no capítulo 2, o motor de corrente contínua pode ser modelado como um sistema de segunda ordem. No projeto do controlador adaptativo, porém, decidiu-se

representar o modelo do motor das juntas do Rhino por uma função de transferência de primeira ordem, como a equação 5.6. Isso só foi possível porque uma das características do controlador adaptativo é fazer com que a saída se comporte de acordo com a saída de referência, mesmo se o modelo real e o modelo de referência adotado sejam diferentes.

A elaboração do experimento é crucial para uma identificação paramétrica bem sucedida. Em problemas de controle, isso se resume no sinal de entrada aplicado ao sistema. No caso de identificação dos parâmetros de um motor, isso consiste na determinação dos coeficientes da função de transferência que descrevem a dinâmica do motor de corrente contínua.

O experimento executado para a obtenção de dados na identificação paramétrica consistiu em aplicar uma tensão que varia de -12 à 12 volts aleatoriamente sobre todos os motores, com um período fixo entre as mudanças de tensão. Os períodos considerados foram de 100 ms, 500 ms, 1000 ms e 2000 ms. Para cada experimento, foram identificados os parâmetros necessários para realizar o controle adaptativo. O sinal de saída do sistema a partir do modelo identificado foi então obtido. Com estes resultados, escolheu-se aquele que melhor representou o sinal de saída e os parâmetros estimados foram inseridos no programa de modo a fazer o controle das juntas do robô Rhino. Os resultados da estimação paramétrica da junta da base é mostrada na figura 5.5.



(a)

$$y(s) = \frac{1338}{s + 6.326} u(s)$$

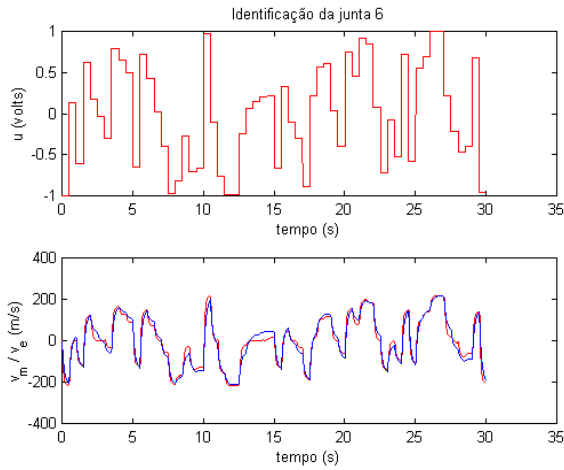
$$\theta_1 = 0.00374928791889$$

$$\theta_2 = -9.786842976045173e - 004$$

$$am = -0.97530991202833$$

$$bm = 0.02469008797167$$

$$\gamma = 8$$



(b)

$$y(s) = \frac{1441}{s + 6.661} u(s)$$

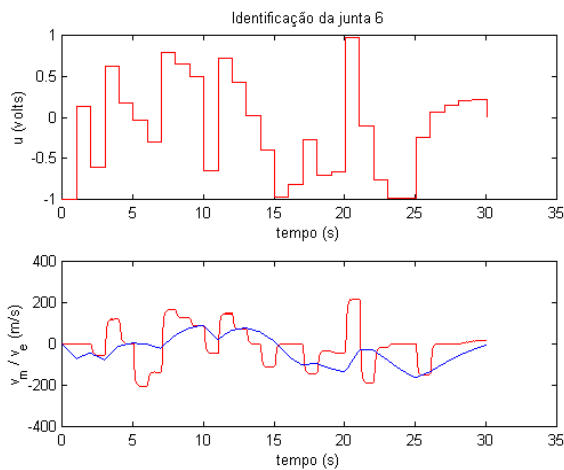
$$\theta_1 = 0.00348473497727$$

$$\theta_2 = -0.00113844237627$$

$$am = -0.97530991202833$$

$$bm = 0.02469008797167$$

$$\gamma = 8$$



(c)

$$y(s) = \frac{83.47}{s + 0.3132} u(s)$$

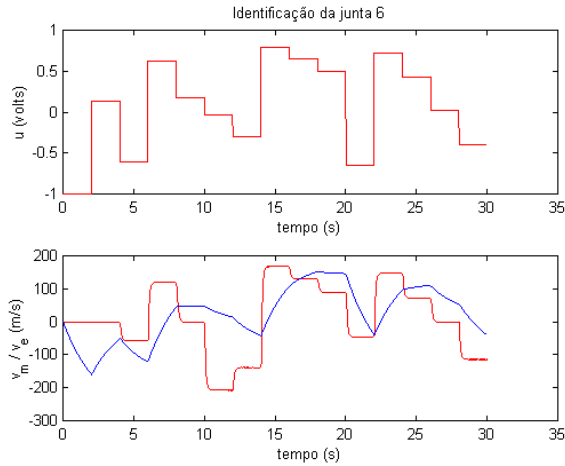
$$\theta_1 = 0.05920583379465$$

$$\theta_2 = 0.05545301887212$$

$$am = -0.97530991202833$$

$$bm = 0.02469008797167$$

$$\gamma = 8$$



(d)

Figura 5.5 _ Identificação da função de transferência da junta da base para o controle adaptativo (MRSA) considerando períodos de (a)100 ms, (b)500 ms, (c) 1000 ms e (d) 2000 ms.

A figura 5.5 mostra os resultados da estimação paramétrica da junta da base do robô Rhino XR-4. Os gráficos mostram a tensão aplicada ao motor, com valores aleatórios e período fixo (gráfico superior) e a velocidade de saída do motor (em vermelho, na figura inferior) e a velocidade estimada a partir dos parâmetros estimados (em azul, na figura inferior). Ao lado do gráfico, estão os parâmetros estimados pela regra do MIT de primeira ordem.

Observa-se que pode-se obter um bom modelo de primeira ordem para o motor de corrente contínua, quando o período em que a tensão sobre os motores foi pequeno (para 100 e 500 milissegundos). O melhor resultado obtido foi o experimento com o período de 100 ms, logo os parâmetros estimados neste experimento foram utilizadas como estimativas iniciais no controlador adaptativo implementado para controlar a junta da base no robô Rhino.

$$y(s) = \frac{117.6}{s + 0.4097} u(s)$$

$$\theta_1 = 0.04204723884724$$

$$\theta_2 = 0.03856226579015$$

$$am = -0.97530991202833$$

$$bm = 0.02469008797167$$

$$\gamma = 8$$

6 Conclusões

O objetivo principal deste trabalho foi o projeto e a implementação de uma nova Arquitetura de Controle para as juntas do robô educacional Rhino XR-4, desde o circuito de acionamento de motores à circuitos de interface com o PC, e algoritmo de controle de motores de corrente contínua.

No capítulo 2 foi apresentado o modelamento matemático de um motor de corrente contínua controlado por armadura e a relação que há entre as diversas variáveis que podem ser controladas, como posição e velocidade angular, corrente e torque.

No capítulo 3 foi apresentado o projeto de circuitos de acionamento de motor DC controlado por armadura. O critério de escolha do circuito de acionamento foi baseado no conhecimento das características dos Pittman a serem controlados e das exigências quanto a confiabilidade e controlabilidade do sistema a ser implementado.

A descrição completa da nova Arquitetura de controle foi apresentada no capítulo 4. Durante a implementação do circuito de Controle, deu-se para perceber a diferença entre projetar o sistema e implementá-lo, uma vez que por causa de ruídos da fonte de alimentação, foi necessário adicionar novos componentes no circuito para minimizar o efeito de ruídos nos sinais.

Uma breve descrição do software desenvolvido para o controle das juntas é mostrado no capítulo 5, assim como os resultados e análises obtidas até o momento, além das perspectivas para desenvolvimento futuro.