

Bruno Vilhena Adorno – Sérgio Amorim Reis

**CONTROLES DE VELOCIDADE E TRAJETÓRIA
PARA O ROBÔ SONIC SHARK E SENSORIAMENTO
POR ULTRA-SOM**

Monografia apresentada ao Curso de Engenharia Mecatrônica da Universidade de Brasília como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação.

Orientador: Geovany Araújo Borges (ENE)

Brasília
Curso de Engenharia Mecatrônica
Departamento de Engenharia Elétrica
Faculdade de Tecnologia
Junho, 2004

Dedicatória

Dedico este trabalho à Luciana, Bianca, Sara, Yan e Yago, pelos momentos de paciência e silêncio que eu precisei ao longo destas semanas.

Sérgio Amorim Reis

Este trabalho é dedicado à minha querida mãe, que sempre acreditou nesse momento, mas infelizmente não pôde esperar para presenciá-lo. Dedico ainda à Polyana, cuja presença é e sempre será indispensável para minha felicidade.

Bruno Vilhena Adôrno

Agradecimentos

Não podemos deixar de exaltar nossa mais respeitosa gratidão ao Professor Geovany Araújo Borges, nosso orientador de projeto, que demonstrou geniosos conhecimentos do assunto e muito nos acrescentou, indo além da técnica e da engenharia, demonstrando sempre atitudes de cortesia e humildade.

Agradecemos aos integrantes da UnBots, que permitiram que o Sonic Shark fosse nossa plataforma mecânica de projeto e viabilizasse a realização deste trabalho.

À Mecajun por ter incentivado e patrocinado a equipe de Guerra de Robôs da Universidade de Brasília.

Agradecemos ainda aos amigos, que sempre nos apoiaram e que de uma forma ou de outra fizeram com que a realização deste projeto fosse possível.

Somos gratos ao professor Guilherme Caribé, por ter nos emprestado a plataforma MANTENA de forma a possibilitar testes preliminares, enquanto o Sonic Shark ainda estava em manutenção.

Agradecimentos	3
Sumário	4
Lista de ilustrações	6
Lista de símbolos e abreviações.....	7
Símbolos	7
Abreviaturas	8
Resumo	9
1 Introdução.....	11
1.1 Guerra de Robôs.....	11
1.2 Sonic Shark	11
1.3 Contextualização	13
1.4 Objetivos do projeto.....	14
2 Revisão Bibliográfica	16
2.1 Medições de Posicionamento Relativo.....	16
2.1.1 Odometria.....	16
2.1.2 Navegação Inercial	19
2.1.3 Acelerômetros	20
2.1.4 Codificador Óptico Incremental	22
2.1.5 Tacômetro de Corrente Contínua	25
2.2 Medidor de Distância Ultra-sônico	26
2.3 Controle de Velocidade.....	32
2.4 Controle de Trajetória	33
3 Desenvolvimento.....	39
3.1 Arquitetura de controle	39
3.2 Codificador Óptico Incremental	41

3.3	Controle de Velocidade.....	43
3.4	Implementação do Controle de Trajetória	44
3.5	Implementação da Odometria.....	46
3.6	Sistema de Navegação	46
3.7	Construção da Célula Ultrasônica	48
4	Avaliação Experimental.....	55
4.1	Resultados Finais.	55
4.1.1	Controle de Velocidade.	55
4.1.2	Controle de Trajetória.	56
4.1.3	Célula Ultra-Sônica.....	57
5	Conclusões.....	58
	Proposta para Trabalhos Futuros	60
	Referências.....	61
6	ANEXOS	63
6.1	Listagem Comentada do Código de Controle.....	63
6.2	Plataforma Sonic Shark.....	68
6.2.1	Estrutura Mecânica.....	68
6.2.2	Projeto Eletro-eletrônico	70
6.2.2.1	PCI Controladora.....	72
6.2.2.2	Módulo Rádio Controle.....	72
6.2.2.3	Módulo Interface	72
6.2.2.4	Módulo de Potência – Ponte H Dupla	74
6.2.2.5	Motor Bosch GPB 12V DC.....	75
6.2.3	Embarque de Controle Remoto	76
6.3	Esquemáticos dos Circuitos Eletrônicos.....	77
6.3.1	Célula Ultra-sônica.....	77
6.3.2	Módulo Auxiliar Multiplexador (CODEC).....	78
6.3.3	Condicionador do Codificador Incremental	79
6.3.4	Circuitos do Sonic Shark.....	80

Lista de ilustrações

Figura 1. Sonic Shark sem carenagem.....	12
Figura 2. Robô inserido no plano X-Y.....	18
Figura 3. Exemplo de um acelerômetro (Adaptado de Adôrno & Baptista, 2004).....	21
Figura 4. Codificador Óptico Incremental (Adaptado do catálogo da Agilent Technologies).....	23
Figura 5. Sinal do Codificador. (a) Sinal atrasado 90°. (b) Sinal adiantado 90°.....	24
Figura 6. Reflexão Sonora.....	26
Figura 7. Incidência da onda sonora.....	27
Figura 8. Processo de medida de distância por ultra-som.....	30
Figura 9. Influência mecânica do sensor.....	31
Figura 10. Robô com tração diferencial e representação das variáveis de erro da trajetória.....	35
Figura 11. Desvio de trajetória para segmentos de reta.....	37
Figura 12. Desvio de trajetória para arcos suaves.....	38
Figura 13. Arquitetura de controle (Adaptado de Borges et al., 2000).....	39
Figura 14. Circuito de condicionamento de sinal (Fonte: Bellardi & Sousa, 2003).....	42
Figura 15. Disco codificador construído para o Sonic Shark.....	42
Figura 16. Foto da Instalação do Codificador Óptico.....	43
Figura 17. Coordenadas polares do ponto de destino da trajetória.....	47
Figura 18. Diagrama de saída do Módulo Ultra-som.....	49
Figura 19. Diagrama de entrada do Módulo Ultra-som.....	49
Figura 20. Tempos envolvidos na medição de distância.....	50
Figura 21. Foto comparativa do módulo Ultra-sônico.....	53
Figura 22. Foto do Módulo de Ultra-som Montado com Dutos Direcionadores.....	53
Figura 23. Foto do módulo auxiliar multiplexador.....	54
Figura 24. Visão superior da estrutura mecânica e seus componentes.....	69
Figura 25. Foto real da estrutura mecânica do Sonic.....	70
Figura 26. Diagrama modular do sistema eletrônico do Sonic Shark.....	71
Figura 27. Foto do Sistema Eletro-eletrônico.....	71
Figura 28. Lógica de proteção e isolamento óptica.....	74
Figura 29. Circuito da ponte H.....	75
Figura 30. Curvas de desempenho do motor Bosch GPB do Sonic Shark.....	75
Figura 31. Esquemático Módulo Ultra-sônico.....	77
Figura 32. Esquemático Módulo Multiplexador.....	78
Figura 33. Esquemático Condicionador do Encoder.....	79
Figura 34. Esquemático PCI Base do Sonic.....	80
Figura 35. Esquemático PCI Base do CW552 e Rádio.....	81
Figura 36. Esquemático PCI Interface do Sonic.....	82
Figura 37. Esquemático PCI Ponte H.....	83
Figura 38. Esquemático PCI Rádio Controle para a HP48.....	84

Lista de símbolos e abreviações

Símbolos

b	: distância entre as rodas de tração, em metros;
c	: velocidade do som variável com a temperatura do ar;
c_i	: velocidade do som variável com a temperatura e direção do ar em relação ao som;
c_o	: velocidade do som variável com a temperatura e direção do ar em relação ao som;
$\vec{e}_i^T \cdot \vec{\omega}$: Escalar resultante do vetor de direção da velocidade do som e da velocidade do ar em uma direção qualquer.
K_i	: Ganho integral dos controladores PI de velocidade dos sistemas de propulsão;
K_p	: Ganho proporcional dos controladores PI de velocidade dos sistemas de propulsão;
K_Γ	: Ganho proporcional relacionado a Γ no controlador de trajetória;
K_ς	: Ganho proporcional relacionado a θ no controlador de trajetória;
L	: distância a ser percorrida pelo robô;
N	: número total de ranhuras no disco codificador;
n_d e n_e	: número de pulsos gerados pelo codificador óptico das rodas direita e esquerda, respectivamente;
P	: ponto de referência de posicionamento do robô;
r	: raio das rodas de tração, em metros;
R	: índice de reflexão acústica;
T_{meio}	: temperatura do meio de propagação (ar) em °C.
v_d e v_e	: velocidades lineares das rodas direita e esquerda, respectivamente, em m/s;
v_p	: velocidade de referência do robô;
(x_d, y_d)	: coordenadas de destino do ponto P no plano cartesiano;
(x_p, y_p)	: coordenadas do ponto P no plano cartesiano;
\dot{x}	: velocidade do ponto P em relação ao eixo x ;
\dot{y}	: velocidade do ponto P em relação ao eixo y ;
α	: ângulo de orientação desejado para o robô;
ϕ	: ângulo de orientação do robô em relação ao eixo x ;

$\dot{\phi}$: velocidade angular do ponto P ;
ϕ	: deslocamento angular da roda;
Γ	: deslocamento ortogonal;
θ	: ângulo de desvio;
ν	: Temperatura do ar em Kelvin;
ω_d e ω_e	: velocidades angulares das rodas de tração direita e esquerda, respectivamente, em rad/s;
ω_d^* e ω_e^*	: velocidades angulares de referência das rodas de tração direita e esquerda, respectivamente, em rad/s;

Abreviaturas

CPR	: Ciclos por revolução (<i>Cycle Per Revolution</i> ou <i>Counts Per Revolution</i>);
CW552	: <i>hardware</i> de controle produzido pela empresa Controlware Automação;
ECHO	: Pulso de retorno de busca;
FM	: Freqüência Modulada;
FSK	: <i>Frequency Shift Keying</i> ;
INIT	: Pulso de início de busca;
PI	: Proporcional Integral
PID	: Proporcional Integral Derivativo
PWM	: Modulação por largura de pulso (<i>Pulse Width Modulation</i>);
Rx	: Recepção via rádio;
Rayles	: Unidade de medida de impedância sonora do meio. Equivale a $\text{kg/m}^2\text{s}$;
Tx	: Transmissão via rádio;
ToF	: Tempo de vôo (<i>Time of Flight</i>).

Resumo: Este projeto consiste no desenvolvimento de um sistema de controles de velocidade e trajetória e sensoriamento por ultra-som para uma plataforma móvel rádio comandada, o robô Sonic Shark. Esta plataforma foi concebida para operar em situações de alto risco de colisões, onde ocasionalmente poderão ser exigidas manobras ágeis em modalidades de ataque e defesa. O Sonic Shark possui configuração diferencial tracionada por dois motores elétricos de corrente contínua independentes, podendo atingir velocidades de até 4 m/s. O controle de velocidade é implementado usando um controlador Proporcional Integral discreto, sendo que este modelo se mostrou inicialmente suficiente para o grau de rejeição a perturbações e erros de regime permanente. Para o controle de trajetória foi utilizado um controlador proporcional. Resultados experimentais ilustram o funcionamento satisfatório do sistema.

Abstract: This project consists on the development of velocity and trajectory controllers for a radio commanded platform named Sonic Shark. A system for obstacle detection based on ultrasound is also developed. This platform has been designed to operate in environments with high risk of collisions, which is the case of Robot's War Competitions. In such cases, the robot should be able to perform maneuvers for attack and defense. Sonic Shark has a differential configuration commanded by two independent D.C. motors. It can reach velocities up to 4m/s. Velocity control is implemented by using a discrete time Proportional Integral controller. This model showed satisfactory performance with respect to the rejection of disturbances and steady state errors. Trajectory control employes a proportional feedback controller. Experimental results illustrate the satisfactory performance of the entire system.

1 Introdução

1.1 Guerra de Robôs

A Guerra de Robôs no Brasil surgiu como um incentivo ao envolvimento dos alunos de Engenharia de Controle e Automação no mundo da robótica e no desenvolvimento de dispositivos mecatrônicos. O objetivo do evento é fazer com que os alunos de mecatrônica possam aprofundar seus conhecimentos em mecânica, eletrônica e controle. Sendo assim, declara-se campeã da competição a equipe que conseguir fazer um protótipo robusto, ágil e que supere os outros robôs devido a sua resistência mecânica.

O nível de conhecimento agregado aos protótipos vem aumentando a cada nova edição do evento dada a troca de informações entre as instituições participantes e devido ao crescente número de alunos interessados em desenvolvimento na área de robótica.

Até o momento houve três edições do evento, sendo que a equipe da Universidade de Brasília participou das duas últimas, tendo sido consagrada com o terceiro lugar, entre dez equipes participantes, na sua última participação em 2003.

1.2 Sonic Shark

O Sonic Shark (Figura 1) foi criado com a finalidade de participar da Guerra de Robôs. Ele possui aproximadamente 50kg e é tracionado por dois motores 12V CC de 0,5cv e 3000rpm cada, acoplados a rodas numa configuração diferencial, podendo atingir velocidades próximas de 4m/s. Ele ainda possui uma roda livre em sua parte anterior, formando então três pontos de apoio entre o robô e o solo.

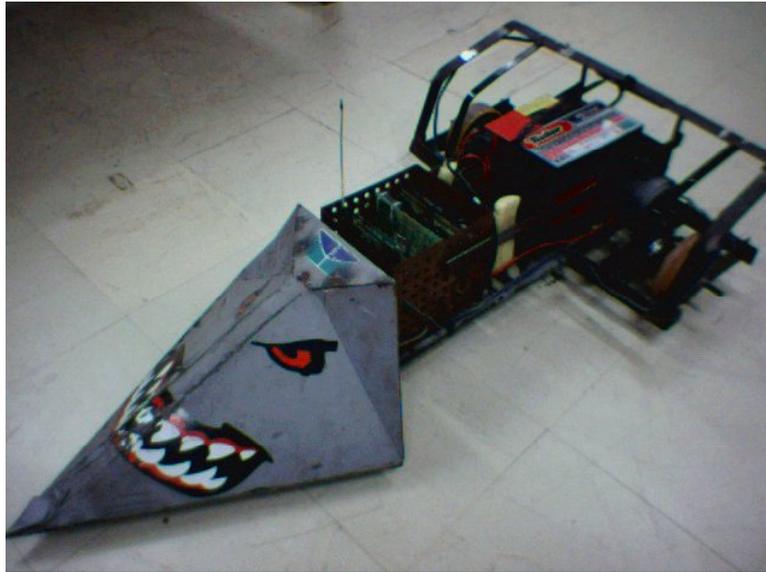


Figura 1. Sonic Shark sem carenagem.

Além disso, o Sonic Shark possui um rádio FM-FSK que possibilita a recepção de comandos, tornando-o em princípio um veículo telecomandado. Sua arma é a rampa que se encontra em sua extremidade anterior. Para que ela seja eficaz, o robô tem que seguir uma trajetória precisa rumo ao seu alvo, de forma a fazer com que este suba na rampa e tombe, ficando impossibilitado de se movimentar.

Inicialmente, o controle de velocidade do Sonic Shark era feito em malha aberta. Dado a pequenas diferenças em seus atuadores, o robô inicialmente não conseguia manter-se em linha reta, de forma a exigir uma maior destreza de quem o estivesse operando. Além disso, seu algoritmo de controle era baseado na recepção de comandos via rádio e execução imediata destes comandos, sem nenhum processamento prévio por parte do robô.

1.3 Contextualização

A robótica vem despertando o fascínio de muitos setores da sociedade e uma das coisas realizadas para atrair a atenção para esta área promissora foi a Guerra de Robôs, um evento motivado pelas universidades afins devido ao potencial de estudos existente nesta atividade. Como toda área de conhecimento que está sendo explorada, os conceitos de robótica e controle aplicados no evento citado estão ficando cada vez mais avançados.

Sendo assim, o robô Sonic Shark vem acompanhando esta evolução. Em sua primeira participação na Guerra de Robôs, falta de robustez e de funcionamento adequado de seu hardware foram os principais problemas encontrados. Estes problemas foram solucionados através da modularização de seu sistema eletro-eletrônico, de forma que o robô foi equipado com um hardware que minimizasse a margem de defeitos.

Para que o sistema pudesse ser controlado, usou-se como hardware de controle a placa CW552 da empresa Controlware, baseada no microcontrolador Phillips 80552 da família Intel 8051. Foram ainda elaborados hardware e software para interpretação de comandos por controle remoto para que houvesse o envio de comandos para o robô. O software embarcado na placa CW552 permitia receber comandos via RS232 e os convertia em ações de movimento e parâmetros de velocidade, onde o controle de velocidade era feito todo em malha aberta.

O rádio receptor foi construído baseado em um módulo da empresa Radiometrix¹, numa taxa de transmissão de dados de até 10Kbps, codificado em FSK (Frequency Shift-Keying), o que permitiu uma comunicação eficiente do operador com o robô.

Todo esse sistema previamente construído já havia se mostrado eficiente e confiável, o que gerou uma motivação em aplicar técnicas mais eficientes de controles de velocidade e de trajetória e ainda um sistema de navegação baseado em leitura de distâncias por ultra-som.

¹ Empresa filiada na Inglaterra

1.4 Objetivos do projeto

Este trabalho gira em torno de uma expectativa final que é uma primeira inserção de técnicas de controle de velocidade, controle de trajetória e ainda um sistema de navegação auxiliado por ultra-som para a plataforma Sonic Shark. Para tanto, seguem abaixo as etapas de desenvolvimento do projeto:

- Implementação de um algoritmo de controle de velocidade em malha fechada:

Este sistema deverá ser controlado por software e ser integrado às demais implementações via um código único de controle embarcado. Baseado em um controlador proporcional-integral, este controle deverá garantir velocidades de referência determinadas para cada eixo.

- Implementação de um codificador óptico incremental (*optical encoder*):

Para medição da velocidade do robô deve ser implementado um codificador óptico incremental em cada eixo de tração. Além disso, deve-se obter o condicionamento adequado do sinal de forma que ele possa ser processado pelo hardware de controle. Este sistema é o nível mais básico que viabilizará o controle de trajetória através de técnica de odometria e de um controle de velocidade.

- Implementação de um algoritmo de controle de trajetória:

Este nível de controle é provido de informações da odometria como posicionamento em uma determinada trajetória e tem como saída as velocidades do sistema de tração diferencial com eixos independentes, para que esta trajetória seja seguida, mesmo utilizando-se apenas de dados da cinemática direta.

- Construção de células ultra-sônicas

Ainda este projeto visa à construção de um sistema de medição de distâncias composto por 6 (seis) células de ultra-som localizadas na periferia da estrutura, de modo a permitir leituras do ambiente externo ao redor do robô.

- Ajuste de desempenho e definição de parâmetros de navegação:

Os controles podem ser fundidos em um nível maior de abstração do sistema. Desta forma, surge a possibilidade de se utilizar técnicas simples de navegação de modo a tornar essencial o levantamento de parâmetros estratégicos para as mais variadas situações de navegação.

2 Revisão Bibliográfica

O problema da navegação em robôs móveis pode ser resumido basicamente em três questões principais: “Onde estou?”, “Para onde vou?” e “Como vou?”. Este capítulo apresenta algumas técnicas e sensores que tentam suprir os sistemas robóticos com informações que possam ser usadas para responder a estas perguntas.

2.1 Medições de Posicionamento Relativo

Em todos os ramos da robótica a localização espacial do robô é fator presente e necessário para o sucesso do sistema. O posicionamento pode ser feito de várias formas e este trabalho discorre sobre as vantagens e desvantagens de alguns métodos. Dentre as mais diversas técnicas de posicionamento, existem as que podem ser usadas para descrever o posicionamento com relação a um marco referencial (posicionamento global ou absoluto), ou com relação à última posição conhecida/estimada (posicionamento relativo).

2.1.1 Odometria

Este método utiliza sensores para estimar a posição da roda e orientação de sua direção. É hoje em dia a técnica mais utilizada no posicionamento de robôs móveis e se encontra na maioria destes (Borenstein et al., 1996). Isso se dá por vários motivos:

- Pode ser fundida a outros métodos de posicionamento de forma a obter dados mais confiáveis;

- Para o caso de navegação referencial usando marcadores (*landmarks*), pode-se usar a odometria para aumentar a precisão do posicionamento entre um marcador e outro. Além disso, caso se tenha um baixo erro odométrico pode-se inclusive diminuir a quantidade de marcadores utilizados;
- Em muitos casos a odometria é a única informação para navegação, quando não se tem nenhuma referência externa ou os outros sensores do sistema não geram dados confiáveis.

Sua principal desvantagem é possuir um erro acumulativo que aumenta com a distância percorrida pelo veículo. Uma maneira de se reduzir este erro é usar uma referência independente para fazer a correção periódica do mesmo.

Os erros de odometria podem ser classificados como sistemáticos ou não-sistemáticos e suas causas podem ser as seguintes (Borenstein et al., 1996):

Erros Sistemáticos:

- Diâmetro desigual das rodas;
- A largura da base da roda é diferente da largura nominal ou então a roda possui irregularidades que fazem mudar seu ponto de apoio;
- Desalinhamento das rodas;
- Resolução e taxa de amostragem finitas do codificador utilizado;

Erros Não-Sistemáticos:

- deslizamento das rodas;
- pisos irregulares;
- passagem por cima de objetos inesperados no solo, etc.

Destas fontes de erros, os erros sistemáticos são os mais fáceis de serem minimizados.

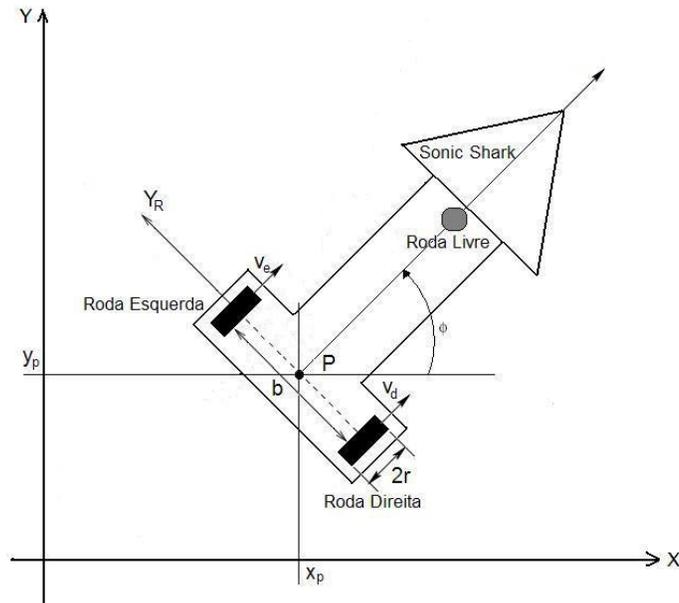


Figura 2. Robô inserido no plano X-Y

As equações de odometria são geralmente derivadas do modelo cinemático direto. Conforme ilustrado na figura 2, dado um robô inserido no plano cartesiano X-Y, movido por tração diferencial com duas rodas de raio r , direita e esquerda, movendo-se com as velocidades $v_d(t)$ e $v_e(t)$, respectivamente. Tem-se que

$$\dot{x}(t) = \frac{(v_e(t) + v_d(t))}{2} \cdot \cos(\phi(t)), \quad (1)$$

$$\dot{y}(t) = \frac{(v_e(t) + v_d(t))}{2} \cdot \text{sen}(\phi(t)), \quad (2)$$

$$\dot{\phi}(t) = \frac{(\omega_d(t) - \omega_e(t)) \cdot r}{b}, \quad (3)$$

com ϕ sendo o ângulo com relação ao eixo X e ω_d e ω_e são as velocidades angular das rodas direita e esquerda, respectivamente. b é o comprimento do eixo imaginário que liga a roda esquerda à direita e P é o ponto médio deste eixo imaginário. \dot{x} e \dot{y} são as velocidades do ponto P em relação ao eixo X e ao eixo Y, respectivamente.

Usando-se o método de discretização de Euler tem-se que:

$$x(k) = x(k-1) + T \cdot \frac{(v_e(k) + v_d(k))}{2} \cdot \cos(\phi(k)), \quad (4)$$

$$y(k) = y(k-1) + T \cdot \frac{(v_e(k) + v_d(k))}{2} \cdot \text{sen}(\phi(k)), \quad (5)$$

$$\phi(k) = \phi(k-1) + T \cdot \frac{(\omega_d(k) - \omega_e(k)) \cdot r}{b} = \phi(k-1) + \frac{T}{b} \cdot (v_d(k) - v_e(k)), \quad (6)$$

com b sendo a distância entre as rodas esquerda e direita, T o período de amostragem, $x(k)$ e $y(k)$ são as coordenadas no plano X - Y e $\phi(k)$ é o ângulo de orientação do robô com relação ao eixo X .

2.1.2 Navegação Inercial

A navegação inercial foi primeiramente usada em larga escala em aviões, tendo sido rapidamente adaptada para o uso em mísseis e em naves espaciais (Borenstein et al., 1996).

Seu princípio de operação consiste em medir as acelerações nos eixos de referência e integrar uma vez os dados obtidos para se obter a velocidade, e integrar mais uma vez para se obter a informação da variação de posição. Este aparato fica em cima de uma plataforma estabilizada por giroscópios, que são usados para estimar a orientação do sistema durante o processo de navegação.

Há vários problemas no emprego da navegação inercial, sendo um deles devido ao fato de os giroscópios, quando influenciados por uma interferência qualquer, fornecem uma medição errada da atitude da plataforma, de forma que os dados obtidos dos acelerômetros tornam-se inconsistentes. Além disso, durante o processo de integração os erros também são integrados, conferindo uma natureza acumulativa ao erro de medição (Borenstein et al., 1996).

No entanto, dado que a navegação inercial não depende de referências externas, em ambientes onde é difícil capturar dados confiáveis, a navegação inercial é um dos poucos meios possíveis de se obter informações de orientação do robô. Resultados obtidos pela Université Montpellier (Vaganay et al., 1993a; 1993b) mostram que o uso da navegação inercial só é recomendado quando é utilizado em conjunto com outros métodos.

2.1.3 Acelerômetros

Um acelerômetro mecânico é um dispositivo sensor que possui três componentes básicos: uma massa, geralmente chamada de massa de prova ou massa sísmica; um arranjo que conecta a massa ao aparato transdutor e um transdutor que fornece os valores de aceleração medidos na forma de grandezas mensuráveis (Adorno & Baptista, 2004).

O princípio básico é o de que uma força agindo em uma massa de prova provoca uma aceleração que pode ser medida por transdutores. Em geral tem-se uma mola com coeficiente de elasticidade conhecido que vai contra-balancear o efeito da força exercida na massa de prova. Através de um transdutor pode-se então obter o valor da aceleração exercida no corpo de prova. Existem vários tipos de acelerômetros, sendo uma das configurações mais básicas a do acelerômetro massa-mola, como mostrado na Figura 3.

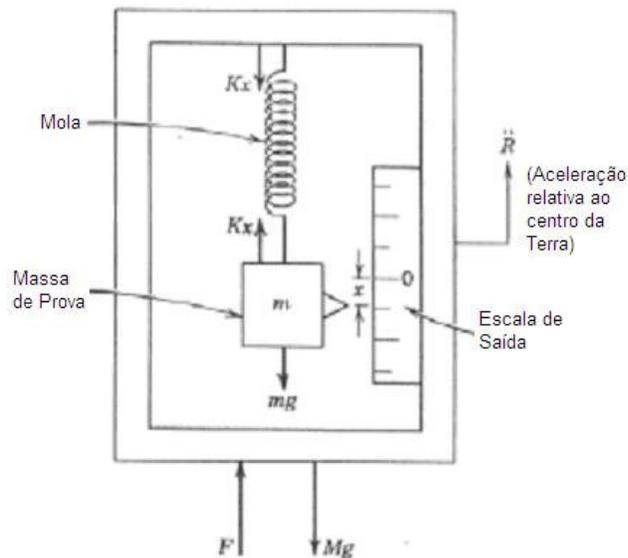


Figura 3. Exemplo de um acelerômetro (Adaptado de Adorno & Baptista, 2004).

A classificação dos diversos acelerômetros se dá devido ao princípio de medição. Existem no mercado acelerômetros que utilizam transdutores compostos por materiais piezoelétricos, ou seja, materiais que quando sujeitos a deformações geram uma diferença de potencial em seus terminais. Transdutores capacitivos também são largamente utilizados, principalmente quando utilizados na configuração diferencial. Há ainda os acelerômetros que são baseados em transdutores resistivos ou magnéticos (Doebelin, 1983).

Em geral os acelerômetros possuem um uso mais restrito na robótica móvel, pois apresentam um baixo desempenho a baixas acelerações (Bellardi & Souza, 2003). No entanto, para medição de acelerações estáticas, tais como a gravidade, os acelerômetros podem ter uma aplicação bastante interessante quando utilizados em fusão a outros métodos estimadores de posicionamento, como, por exemplo, a odometria. Em uma situação em que o solo não seja regular, por exemplo, e sim composto de subidas e descidas, a planta baixa gerada somente pelo sistema de odometria estaria incorreta, pois este método parte do princípio que o solo é regular e plano. Fazendo-se a fusão da odometria com a medição da inclinação usando um

acelerômetro, pode-se construir uma planta baixa mais precisa e obter um posicionamento tridimensional do robô.

2.1.4 Codificador Óptico Incremental

Para realização de odometria em robôs móveis, podem-se utilizar vários tipos de sensores. Um dos mais utilizados é o codificador óptico incremental, que através de um disco codificador, possibilita a realização de medições de incrementos angulares dos eixos das rodas.

Em geral sua grande utilização em robótica móvel se justifica pelo fato de ter uma boa imunidade a ruídos, ser relativamente simples, ter um tamanho reduzido, de ter uma ampla bibliografia que o descreve, além de ser facilmente encontrado no mercado.

Para o uso do codificador óptico incremental são necessários dois elementos básicos para seu funcionamento. Um cria as condições para medidas angulares, o chamado disco codificador. Outro gera um sinal luminoso e o capta em outra extremidade, o codificador óptico incremental propriamente dito. Os dois juntos atuam como um sistema de chaves e se baseiam no princípio da interrupção luminosa para geração de sinais em acordo com o movimento angular que se quer medir.

No contexto deste projeto será especificado um codificador circular aplicado para a odometria, conforme mostrado na Figura 4. A resolução do sistema de odometria depende basicamente do número de furos ou janelas presentes no disco codificador, onde uma revolução no disco corresponde a 360° do eixo, que corresponde por sua vez a 360° informados ao sistema elétrico. A essa especificação dá-se o nome de CPR (Cycle Per Revolution) ou ciclos por rotação, que é a quantidade de janelas do disco.

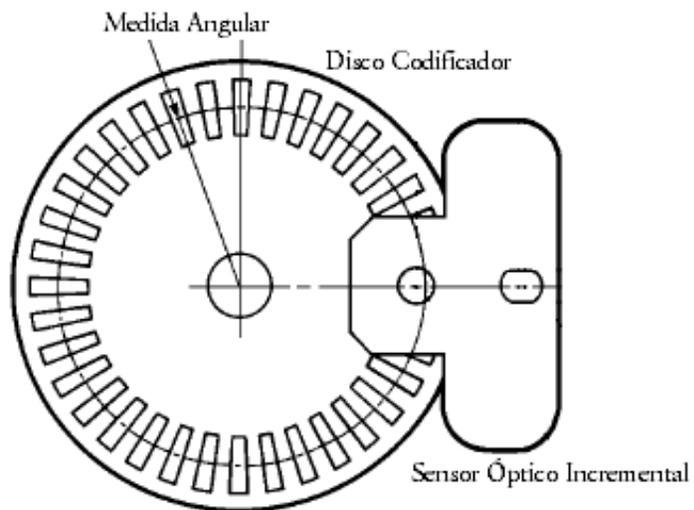


Figura 4. Codificador Óptico Incremental (Adaptado do catálogo da Agilent Technologies).

A informação de interesse da saída do codificador é a frequência dos pulsos gerados pela velocidade de rotação do eixo medido. Quanto maior a velocidade, maior a frequência e vice-versa. Este sinal é processado de forma a obter a contagem dos pulsos para cada ângulo ou rotação.

Um tipo de codificador óptico bastante popular é o de canal simples (*tachometer encoder*). Este sensor trabalha muito bem em aplicações onde se têm velocidades médias e altas. Porém, em velocidades extremamente baixas passa-se a ter problemas com ruídos e estabilidade dado aos erros de quantização ((Nickson, 1985) – referenciado em Borenstein et al., 1996). Além disso, com somente um canal não é possível medir o sentido de rotação do disco.

Para resolver o problema da medição do sentido de rotação foi criado o codificador de canal duplo (*Phase-quadrature incremental encoders*), onde o sinal de um canal está defasado do sinal do outro canal em 90°. Sendo assim, como existem dois sinais, um pode ser usado como referência e o outro para determinar o sentido, como se pode ver nas figuras 5 (a) e 5 (b). Pode-se também, usando este tipo de codificador óptico, usar os dois

sinais para medir velocidade e deslocamento simultaneamente, tendo ainda a vantagem de aumentar a resolução da medição.

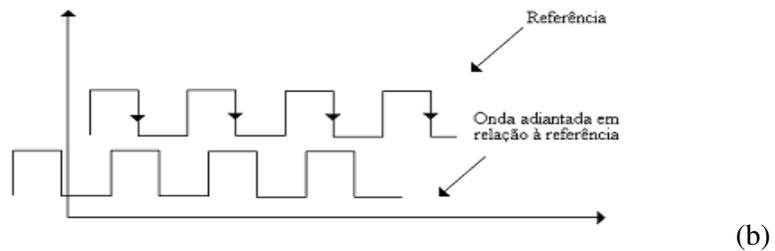
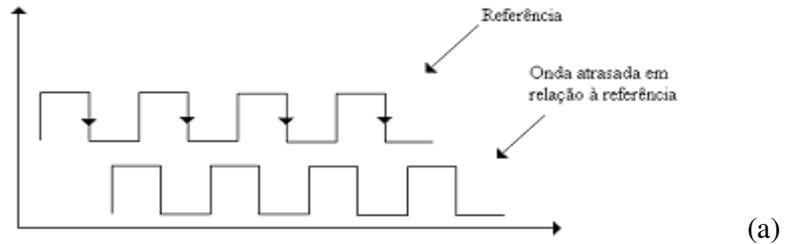


Figura 5. Sinal do Codificador. (a) Sinal atrasado 90°. (b) Sinal adiantado 90°.

Um dos problemas encontrados ao utilizar codificadores ópticos é o *jitter*. *Jitter* é o desvio de uma transição de sinal de sua posição ideal no tempo. Geralmente é causado por imperfeições na medição e tem seus efeitos agravados com o aumento da frequência. Isto porque, para um aumento da frequência, tem-se que o período de tempo de um bit torna-se cada vez menor, sendo que o erro ocasionado pelo *jitter* torna-se proporcionalmente maior, podendo até causar a corrupção dos dados.

O efeito do *jitter* é agravado quando se tem um disco codificador que não é plano e também não é rígido o suficiente, se comparado à resolução de medição. Sendo assim, para minimizar este efeito é aconselhável a utilização de um disco o quanto mais plano possível e feito de um material rígido o suficiente de forma a não dobrar facilmente.

2.1.5 Tacômetro de Corrente Contínua

Este tipo de dispositivo tem seu uso bastante comum entre as formas de medição de velocidade de um eixo. Dentre os mais usados tipos de tacômetro encontram-se os tacômetros de corrente contínua (*DC Tachometer*).

Seu funcionamento se baseia em um simples gerador de corrente contínua que produz uma tensão de saída proporcional à velocidade de seu eixo. Pode ser encontrado de duas formas: com ímã permanente ou com campo excitado. Apesar de seu princípio de funcionamento ser igual ao de um gerador, sua fabricação tem um projeto que leva em conta que não será usado para gerar energia e sim gerar um sinal para medida de uma grandeza mecânica. Neste sentido ele é também considerado um transdutor. Portanto, seu princípio básico de operação é descrito pela equação (7) (Doebelin, 1983), estando acoplado a um eixo para medida rotacional.

$$e_o = \frac{n_p n_c \phi N}{60 n_{pp}} 10^{-8}, \quad (7)$$

com e_o sendo a tensão média de saída, n_p é o número de pólos, n_c é o número de condutores na armadura, ϕ é o fluxo magnético por pólo, N é a velocidade em RPM e n_{pp} é o número de elos paralelos entre a escova do terminal negativo e positivo. Verifica-se também que a velocidade e_o inverte a polaridade quando o sentido da velocidade angular também inverte.

Devido ao número finito de enrolamentos o sinal de saída contém uma pequena componente alternada (*ripple*). Um filtro passa-baixas pode ser eficiente para reduzir esta componente para altas velocidades mas não o seria para velocidades menores.

Um tacômetro típico teria uma sensibilidade de 7V por 1000RPM, uma velocidade máxima de 5000RPM, com uma não-linearidade de 0,07% em uma faixa de 0 a 3600RPM e 2% de *ripple* a uma velocidade de 100RPM. Estes são parâmetros considerados bons para um tacômetro de pequeno porte (Doebelin, 1983).

2.2 Medidor de Distância Ultra-sônico

O ultra-som, dentre muitas outras aplicações, pode ser um ótimo meio de medição de distâncias. A seguir é descrito como isto pode ser implementado, como funciona e quais as vantagens proporcionadas em relação a outros métodos possíveis.

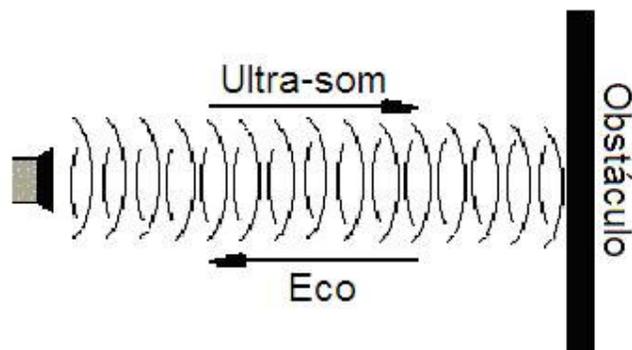


Figura 6. Reflexão Sonora

Um sistema de medição de distância por ultrassom baseia-se no seguinte princípio: um feixe ultra-sônico é dirigido a um obstáculo distante (alguns metros) e as vibrações sonoras, ao encontrar este obstáculo, serão refletidas, retornando uma parte de sua energia em direção à fonte emissora (Figura 6).

Um transdutor receptor recebe este sinal refletido e o entrega a um sistema eletrônico que amplifica e filtra este sinal. Feito isso, a obtenção da distância é feita pela estimação do tempo de vôo (*ToF – Time of Flight*) decorrido entre a emissão e a recepção, levando em consideração as diversas variáveis envolvidas neste processo físico, dentre elas a reflexão, o ângulo de incidência, a diferença de impedância sonora entre o ar e o obstáculo e a temperatura do ar (Borenstein et al., 1996).

A melhor reflexão do feixe para a medida da distância é a que ocorre a um ângulo de 90° (incidência normal) entre a direção do feixe de saída e a superfície. Desta forma, um único sensor acoplado a um robô possibilita medir distâncias corretas em apenas uma única direção.

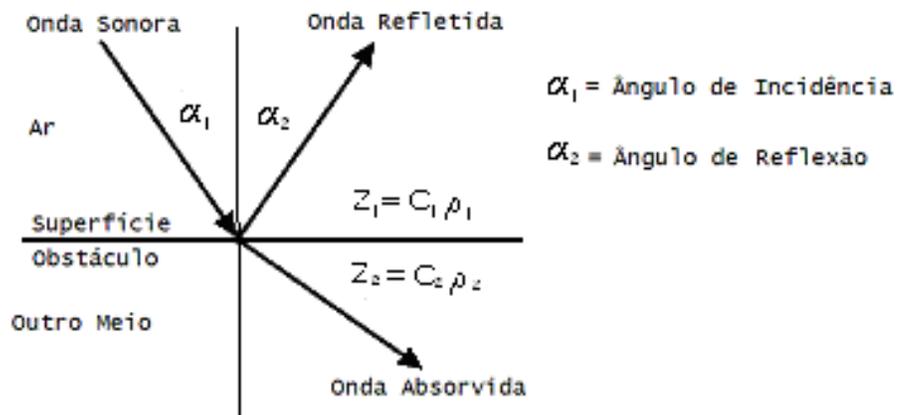


Figura 7. Incidência da onda sonora.

A Figura 7 mostra os aspectos envolvidos na incidência da onda em uma superfície que separa dois meios de propagação. Os mais importantes para este projeto são o ângulo de incidência e o índice de reflexão. A temperatura do ar afeta a precisão da medida, mas mais adiante é possível inferir este grau de influência.

Tanto as características de propagação no ar quanto as do material do obstáculo também são de suma importância para a eficácia deste método de medição, uma vez que o meio por onde se propaga a onda sonora oferece uma certa resistência à vibração sonora (impedância Z). No entanto, esta variável servirá apenas para estimar as diferenças entre o ar e o obstáculo, uma vez que esta diferença é determinante do quanto de energia será refletida pelos variados tipos de obstáculos para que seja possível detectá-los (Briggs & Moir, 1963).

Para ocorrer uma reflexão que satisfaça o objetivo de medição proposto, deve-se considerar o índice de reflexão do material do obstáculo. Este deve sempre tender a aproximar-se da unidade ($R \rightarrow 1$), mostrando que o ar e o material detectado são de diferentes densidades e, portanto melhor desempenho será obtido na detecção e conseqüentemente na medição. Pode-se mostrar pelas teorias da acústica que quanto maior a diferença de impedância entre os meios, mais próximo da unidade se aproxima o coeficiente de reflexão R , entretanto, este não é o tema tratado. A impedância z é calculada conforme a equação simples $z = c \cdot \rho$ que mostra a dependência com a densidade (ρ) do meio ou fluido e a velocidade de propagação (c). É medida em Rayles (1 Rayles = $1\text{kg/m}^2\text{s}$) (Briggs & Moir, 1963).

Pelo princípio de conservação da energia pode ser deduzido que uma parte da energia sonora incidida é refletida, outra é absorvida e o restante é dissipado em forma de calor. Neste trabalho não será necessário considerar a porção dissipada, ou seja, é desprezível (sem perdas) para o sistema de reflexão considerado. Isto significa que a amplitude da onda refletida será dependente apenas do índice de reflexão (Briggs & Moir, 1963).

Este índice, segundo a teoria acústica, varia entre -1 e +1. Nesta faixa existem quatro tipos de reflexão diferentes. As duas mais importantes dizem respeito ao melhor e ao pior caso para o projeto do medidor por ultra-som. São eles:

- Para a impedância do obstáculo (z_2) muito maior que a do ar (z_1), ($R \rightarrow 1$). Significa uma superfície muito rígida, isto é, a maioria da energia acústica será refletida. Neste caso a fase da frente de onda é pouco modificada, não havendo portanto a diminuição da amplitude da onda.
- Para a impedância do obstáculo (z_2) próxima a do ar (z_1), ($R \rightarrow 0$): superfície macia, a maior parte da energia não será refletida ou refletida com muita atenuação.

Para cada material do obstáculo encontrado, o sinal refletido vai ser melhor detectado quanto mais sólido for o material de sua superfície. Para diferentes materiais haverá a possibilidade do sinal de eco não conseguir excitar o receptor. Desta forma, não se pode

esperar que a medição vá detectar qualquer tipo de corpo em sua volta, mesmo este sendo de tamanho suficiente para reflexão (Briggs & Moir, 1963).

Sabe-se que as ondas sonoras propagam no ar a uma velocidade média de 343,8m/s a 25°C. Esta velocidade sofre alterações importantes com a variação da temperatura do ar. Por exemplo, a 40°C a velocidade sobe para 354m/s, conforme a equação (9). A umidade do ar também está no rol das características importantes para a medição ultra-sônica, mas seu efeito não será abordado neste trabalho. Na prática, o ar é uma mistura de gases e esta composição também sofre alterações em função da altitude. Assim sendo, a exatidão da medida deveria considerar todos estes fatores que alteram a velocidade do som no meio. Porém, antes de qualquer coisa, deve ser considerada a precisão que vai ser exigida do sistema ultra-sônico e assim estimar o quanto a variação destas variáveis vão interferir no sistema de medida. A variação da temperatura é considerada neste projeto como a mais crítica. Assim o trato a este aspecto merece um breve aprofundamento.

A temperatura versus a velocidade de propagação é mais precisamente relacionada de acordo com a equação (8) (Wehn & Bélanger, 1997).

$$c_i = c_0 \sqrt{1 + \frac{v}{273^\circ C} + \vec{e}_i^T \cdot \vec{\omega}}, \quad (8)$$

com v sendo a temperatura do ar a velocidade de $\vec{\omega}$. E c_i é a velocidade do som na direção e_i e $c_0 = 331,4m/s$.

Entretanto, é possível verificar experimentalmente que esta equação pode ser simplificada para a equação (9) sem comprometer o resultado final, no caso de medições que não necessitem de alta precisão:

$$c = 331,1 + 0,61 \cdot T_{meio} (^\circ C). \quad (9)$$

O ar mostra também uma forte componente de atenuação, o chamado amortecimento mecânico (*damping*). Em função das características do sinal de ultra-som, ele poderá sofrer maior ou menor amortecimento. Assim é tido que frequências mais altas são mais amortecidas do que as frequências mais baixas. Portanto, para cada aplicação, um

sistema de medição por ultra-som exigirá uma frequência adequada. Frequências da ordem de 20kHz a 40kHz podem ser consideradas "baixas" e sofrem pouco amortecimento no ar, enquanto que frequências de 40kHz a 100kHz, consideradas "altas", sofrem um amortecimento bem maior, porém permite maior resolução da leitura. Para a medição de distâncias neste projeto é aconselhado os 40kHz, até porque também os transdutores comerciais nesta faixa são mais acessíveis.

A emissão para medição de distância deve ser feita de modo pulsado com tempos precisamente determinados (Figura 8). Os sinais envolvidos no sistema de transmissão e eco são gerados e detectados por um sistema de processamento que deverá estar ajustado para operar e calcular com os tempos envolvidos no sistema ultra-sônico como um todo. A Figura 7 apresenta o tipo de processo necessário à leitura do eco.

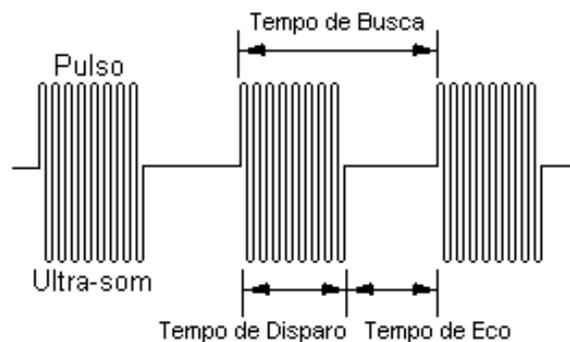


Figura 8. Processo de medida de distância por ultra-som.

No eco as vibrações mecânicas atingirão o transdutor receptor fazendo-a vibrar na frequência de 40kHz. Entretanto, sendo o sensor um dispositivo mecânico, ele possui uma massa e, portanto uma inércia. A massa do transdutor que estava parada vai sendo acelerada até entrar em ressonância, sendo este ponto o de maior amplitude. Quando o pulso termina, isto é, passado o tempo de disparo, a mecânica do sensor que está em amplitude máxima ainda continua vibrando mesmo sem o estímulo. Esta característica é ilustrada na Figura 9 que mostra também que o ponto máximo do eco é atingido sempre na borda final do pulso emitido. Em testes feitos em bancada foi verificado claramente este fenômeno. Uma medição

mais precisa deve considerar sempre esta característica de referência no tempo. Mas esta parcela de tempo é muito pequena e será desprezível para este projeto.

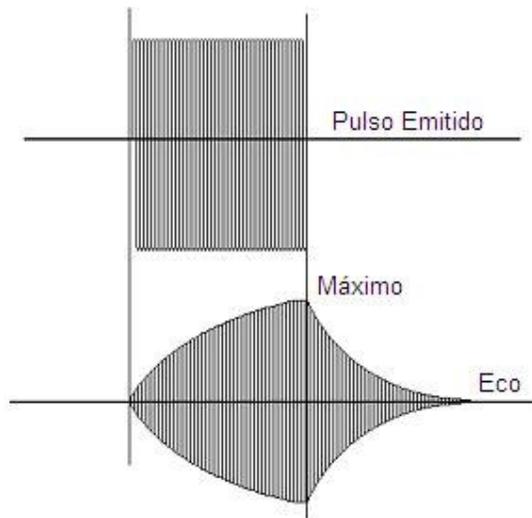


Figura 9. Influência mecânica do sensor.

Este sinal de eco deve ser tratado de forma a tornar-se o mais próximo possível de uma onda quadrada. Este sinal tratado deve ser capaz de excitar uma entrada digital de tal forma a permitir a sua detecção e por fim o cálculo do tempo de vôo.

A aplicabilidade do ultra-som é fator de interesse para a escolha dos sensores, pois são oferecidos vários tipos no mercado, de acordo com o material de sua construção. Estas diferenças influem em suas características de alcance e resistência mecânica. Os tipos mais comuns são os piezoelétricos, mas não são recomendados para o caso de ambientes de vibrações extremas e uma característica indesejável é o seu curto alcance, na faixa de 5mm a 800mm, segundo os catálogos encontrados atualmente no mercado. Transdutores Murata da série MA40 [MURATA] são bastante difundidos no campo da robótica, pois são de baixo custo e boa sensibilidade. Seu alcance varia de 0,2m a 4m numa resolução de até 9mm.

Os problemas mais enfrentados neste tipo de sensoriamento baseado no tempo de viagem de um sinal (ToF) são os seguintes (Borenstein et al., 1996):

- Variações na velocidade de propagação, conforme já citado, promovidas pela inconstância da temperatura do meio (ar) e também da umidade;
- Incertezas na determinação exata do tempo de eco devido a erros sistemáticos do sistema acústico variante no tempo;
- Erros e ruídos inseridos pelos próprios circuitos de tratamento dos sinais medidos;
- Interações indesejáveis geradas por fatores de incidências da onda sonora sobre os diversos tipos de obstáculos e superfícies. Exemplo: cantos, reflexões secundárias (*Crosstalk Path*), etc.

Porém existem vantagens quando na aplicação neste trabalho:

- Precisão final suficiente para medidas pequenas em relação às dimensões da plataforma Sonic e ao sistema de controle proposto.
- Baixo custo de construção por essa acuidade relativamente satisfatória

Desta forma conclui-se que este método de medição de distâncias pode ser bastante útil para obter bons resultados em sistemas de navegação.

2.3 Controle de Velocidade

Existem várias abordagens para o controle de velocidade, que vão desde a abordagem do controlador proporcional até abordagens mais complexas, tais como o controle adaptativo. Contudo, o objetivo aqui não é fazer um estudo sobre qual estratégia é a melhor, e sim apresentar uma estratégia que garanta um bom tempo de resposta e erro nulo em regime permanente.

Uma estratégia que atende a essas especificações é o controlador Proporcional-Integral, pois o canal proporcional se encarrega de diminuir o tempo de resposta do sistema e o canal integral faz com que o erro em regime permanente seja nulo. Este controlador pode ser modelado de acordo com a equação (10).

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau . \quad (10)$$

Derivando-se os dois lados da igualdade, tem-se que

$$\dot{u}(t) = K_p \dot{e}(t) + K_i e(t) . \quad (11)$$

Discretizando (11) e usando o método de Euler tem-se que

$$u(k) = u(k-1) + (K_p + T \cdot K_i) e(k) - K_p e(k-1) . \quad (12)$$

Este modelo discreto do controlador PI tem a vantagem de ser pouco oneroso em termos de cálculos computacionais, podendo ser implementado em plataformas de baixo desempenho.

2.4 Controle de Trajetória

Nas últimas décadas foram propostas várias técnicas de controle de trajetória para veículos autônomos e robôs. Sabe-se que em geral o sistema de controle de velocidade é decisivo para o bom desempenho e para a estabilidade do sistema de controle de trajetória. Já está bem consolidado na literatura, que controladores PID não garantem a estabilidade para veículos não-holonômicos, dentre os quais estão os robôs de tração diferencial.

No entanto, consegue-se obter resultados satisfatórios quando se trata de trajetórias suaves tais como linhas retas e arcos suaves ((Cox, 1991) – referenciado em Borges et al., 2000). Para trajetórias arbitrárias, devem ser usadas técnicas de controle não-linear. Em

geral são utilizados o controle adaptativo, lógica *fuzzy* ou ainda a realimentação não linear de estados.

Existem duas abordagens para o rastreamento de trajetória: a abordagem do controle seqüencial de postura, onde a cada período de amostragem a postura do robô é definida como um ponto sobre a trajetória e ainda há a abordagem do rastreamento de caminho, onde a consideração recai sobre as variáveis que definem o erro de trajetória, que são função da posição atual e da estrutura geométrica do robô.

Será exposto aqui um modelo de controle de trajetória proporcional (Borges et al., 2003) para robôs baseados em tração diferencial. Este modelo baseia-se apenas no modelo cinemático do veículo a ser controlado, de forma que as variáveis dinâmicas são consideradas perturbações e devem ser corrigidas pelo controlador.

O objetivo deste controlador é levar o desvio do robô em relação à trajetória para zero. Esse desvio é definido em termos de uma posição linear e um desvio angular da trajetória. Ele assume que as rodas estão independentes e a dinâmica do veículo não está sendo considerada. Além disso, o controle de cada roda é feito individualmente.

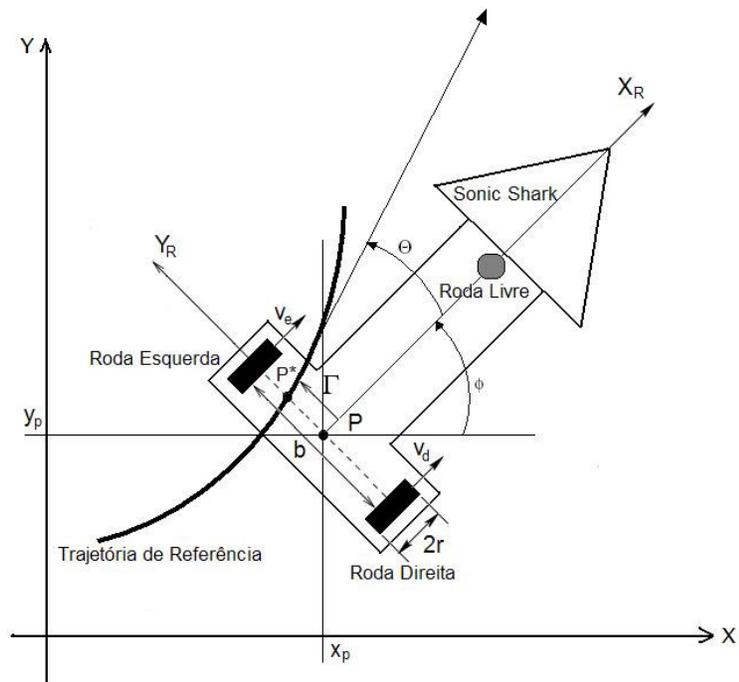


Figura 10. Robô com tração diferencial e representação das variáveis de erro da trajetória

Assumindo que o robô está seguindo a trajetória suave da Figura 10 e ainda que o sistema de velocidade e a odometria estão consistentes, pode-se dizer que (Borges et al., 2000), dado um ponto P que se encontra no meio do eixo imaginário que conecta a roda esquerda à roda direita do robô, tem-se uma reta que é ortogonal a este eixo imaginário e que passa por P . Esta reta faz um ângulo ϕ com o eixo x e um ângulo θ com a tangente da trajetória no ponto P^* . Este ponto é a projeção do ponto P na trajetória. A distância P^*P é chamada de Γ e é uma das variáveis a ser controladas, assim como o ângulo θ . Sendo assim, o objetivo deste controlador é minimizar o valor de Γ e θ de forma que o robô siga a trajetória proposta.

Sabe-se que para obter uma velocidade constante em uma trajetória, as velocidades angular da roda esquerda e direita, ω_e e ω_d , são dadas por (Borges et al., 2000)

$$\omega_e = \frac{v_p}{r} - \Delta\omega(\Gamma(t), \theta(t)) \quad (13)$$

e

$$\omega_d = \frac{v_p}{r} + \Delta\omega(\Gamma(t), \theta(t)).$$

Os parâmetros do controlador são dados por K_Γ e K_θ , conforme as equações (14) e (15). O Γ_c e o R são os parâmetros que dizem quais serão o erro máximo em regime permanente e o raio de trajetória, respectivamente.

$$K_\Gamma = \frac{2v_p b}{r \cdot 4\Gamma_c (R + \Gamma_c)}, \quad (14)$$

$$K_\theta = \sqrt{\frac{2v_p b \cdot K_\Gamma}{r}}. \quad (15)$$

Sendo assim, o valor do $\Delta\omega$ é dado por (15):

$$\Delta\omega = K_\Gamma \cdot \Gamma + K_\theta \cdot \theta. \quad (16)$$

Pelas equações (13) pode-se dizer que as velocidades linear das rodas esquerda e direita, v_e e v_d , são dadas por (17) e (18):

$$v_d = v_p + r\Delta\omega, \quad (17)$$

$$v_e = v_p - r\Delta\omega. \quad (18)$$

Nota-se que este controlador tem a propriedade de manter a velocidade do ponto P constante, propriedade fundamental em aplicações onde se deve ter o transporte de objetos sensíveis a acelerações bruscas.

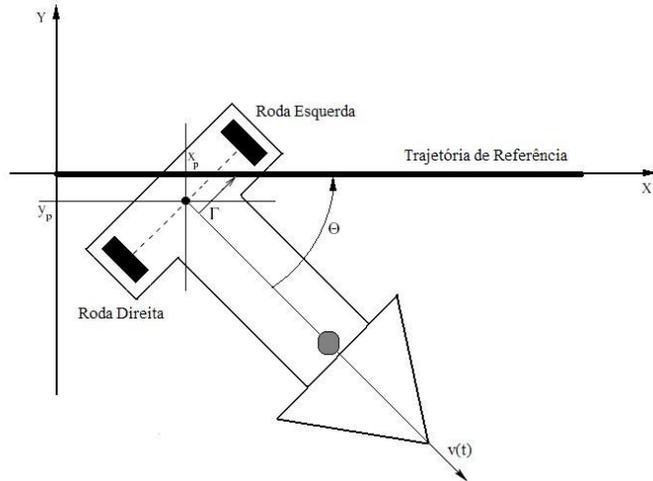


Figura 11. Desvio de trajetória para segmentos de reta.

Dada a Figura 11, onde o que se deseja é levar o robô para uma trajetória reta, tem-se que as condições iniciais do problema são:

$$\begin{aligned} x_p &= 0, \\ y_p &= 0, \\ \theta(t) &= -\phi(t). \end{aligned} \tag{19}$$

A variável Γ a ser minimizada é dada por

$$\Gamma(t) = \frac{y_p}{\cos(\phi(t))}. \tag{20}$$

Neste caso, o ângulo ϕ é dado pelo sensor de deslocamento, que no caso específico do robô Sonic Shark é dado pela odometria. Uma propriedade importante é que para retas, este controlador tem erro nulo em regime permanente.

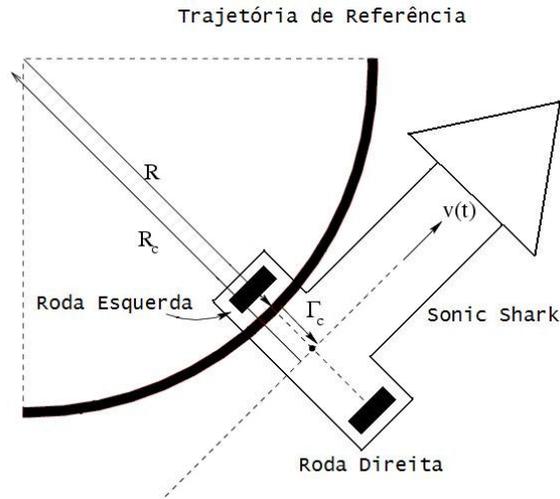


Figura 12. Desvio de trajetória para arcos suaves.

Sendo uma trajetória dada por um arco de circunferência, tal como na Figura 12, assume-se que o robô tenha entrado em regime permanente e tenha apenas um erro residual dado por Γ_c , cuja relação com o raio de trajetória é dada por (21):

$$\Gamma_c = \frac{v_p}{\Delta\omega} \cdot \frac{b}{2r} - R. \quad (21)$$

Este controlador proporcional de trajetória tem a vantagem de ser relativamente simples, ter um bom desempenho e ainda ter um custo computacional pequeno, podendo ser implementado em microcontroladores de baixo custo.

Para trajetórias virtuais, pode-se aproximar qualquer curva suave por uma soma de segmentos de retas, de forma que pode ser utilizada uma estratégia de controle de trajetória usando somente segmentos de retas. Neste ponto, pode-se notar que o controlador garante que o robô sempre convergirá para a trajetória e terá um erro nulo em regime permanente para o caso de retas.

Entretanto, caso ocorra um problema na medição das variáveis Γ_c e θ , o controlador não conseguirá manter o robô na trajetória desejada, mostrando a importância de se ter um subsistema de controle de velocidade e odometria bem ajustados, para o caso do robô Sonic Shark.

3 Desenvolvimento

3.1 Arquitetura de controle

Para o robô Sonic Shark foi desenvolvida uma arquitetura hierárquica de controle, como pode ser observado na Figura 13.

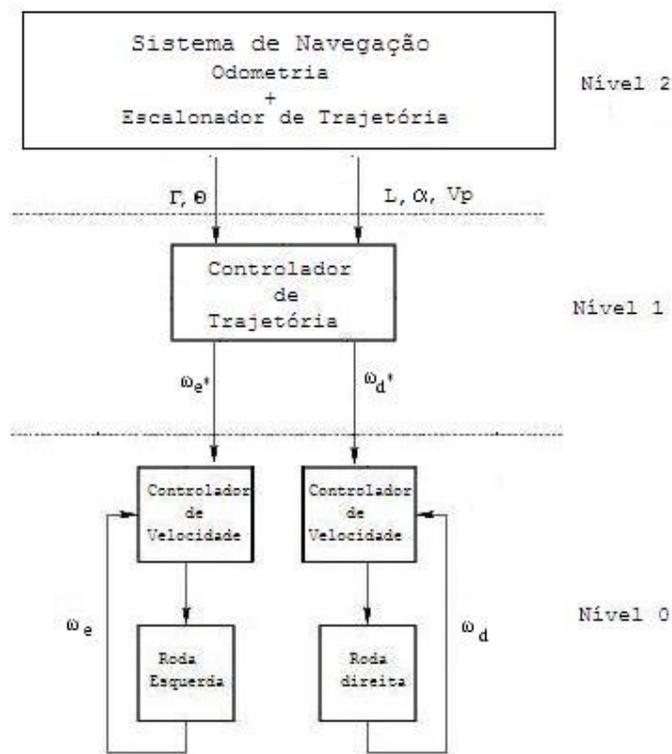


Figura 13. Arquitetura de controle (Adaptado de Borges et al., 2000).

No nível mais baixo está o controle de velocidade. No caso deste trabalho, foi utilizado um controlador PI, que apresentou desempenho satisfatório por minimizar perturbações e fazer o controle de velocidade dos atuadores com um bom tempo de resposta e um erro nulo em regime permanente.

Como o sensor utilizado para realizar controle de velocidade e odometria foi um codificador óptico, optou-se por usar como sinal de referência a frequência f em pulsos/segundo para medição indireta das velocidades angular e portanto linear das rodas do robô. Isto porque, tendo uma base de tempo constante T , pode-se medir as velocidades angular e linear de uma roda através da relação dada por (23) e (24):

$$\Delta\varphi = \frac{n}{N} \cdot 2\pi, \quad (22)$$

$$\omega = \frac{\Delta\varphi}{T} = \frac{n \cdot 2\pi}{T \cdot N}, \quad (23)$$

$$v = \omega \cdot r = \frac{n \cdot 2\pi}{T \cdot N} \cdot r. \quad (24)$$

em que n é o número de pulsos gerados pelo codificador óptico, T é o período de amostragem, N é o número total de ranhuras existentes no disco codificador, r é o raio da roda, v é a velocidade linear da roda, φ é o deslocamento angular da roda e ω é a velocidade angular da roda.

Sendo assim, o nível 0 gera informações sobre o deslocamento angular de cada roda através dos pulsos gerados pelo codificador óptico correspondente n_e e n_d , de forma a obter a localização do robô no sistema de coordenadas X-Y e sua respectiva orientação ϕ . Após fazer o processamento desses dados, o nível 2 fornece ao nível 1 informações sobre o deslocamento linear ortogonal Γ e o desvio angular θ .

O nível 1, com base nas variáveis de erro de trajetória Γ e θ , realimenta o controlador de velocidade com novas velocidades de referência, de forma a fazer com que o robô convirja para a trajetória e mantenha a velocidade no ponto P constante, conforme ilustra a Figura 13.

O nível 2, com base nos dados da odometria, determina trajetórias através dos parâmetros passados para o nível 1, por meio da velocidade do ponto P e das coordenadas polares dos pontos x_d e y_d de destino desejados, que são o comprimento do segmento de reta a ser percorrido L e o ângulo de orientação α , conforme pode ser visto na Figura 17.

O uso de referências indiretas de velocidade angular e linear dadas pela frequência em pulsos/segundo elimina alguns cálculos indiretos, o que diminui o custo computacional dos algoritmos, tornando-os implementáveis em hardware de baixo custo e baixo poder computacional, como no caso do Sonic Shark.

3.2 Codificador Óptico Incremental

Para a realização da odometria utilizou-se um codificador óptico de baixo custo, construído em laboratório. Internamente ele é composto de um diodo emissor de luz (LED) e de um fototransistor ligados em Vcc comum. Sua característica é de um filtro passa-baixas, de forma que à medida que a frequência de pulsos vai aumentando, o sinal vai ficando com sua amplitude mais atenuada.

Para resolver este problema, o sinal de saída do codificador óptico foi ligado à entrada não-inversora de um comparador de tensão LM311. Já a entrada inversora foi ligada a um divisor de tensão, de forma a ter em sua entrada uma tensão de referência de 200mV. Sendo assim, para um sinal acima da tensão de referência, o comparador envia nível lógico 1 para a saída.

Como o LM311 foi alimentado por 0 e 5V, o sinal de saída foi uma onda quadrada, porém com uma amplitude menor que a desejada, que idealmente deveria ser na compatibilidade TTL. Sendo assim, este sinal foi passado por um *Schmidt Trigger* 74LS14 de forma a ser normalizado em 0 e 5V TTL. O circuito de condicionamento está ilustrado na Figura 14.

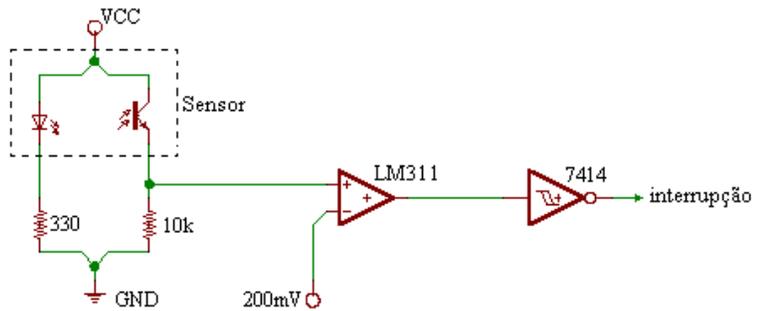


Figura 14. Circuito de condicionamento de sinal (Fonte: Bellardi & Sousa, 2003)

O disco codificador usado foi impresso em papel transparente em dupla camada e tem 100 janelas por revolução, conforme pode ser visto na Figura 15, onde $N=100$. Pelo fato do diâmetro do disco ser muito grande e ser feito de plástico, sua oscilação tornou-se significativa, aumentando o efeito do *jitter*. Porém, com um bom posicionamento do sensor, este efeito foi minimizado de forma que não prejudicasse a odometria.

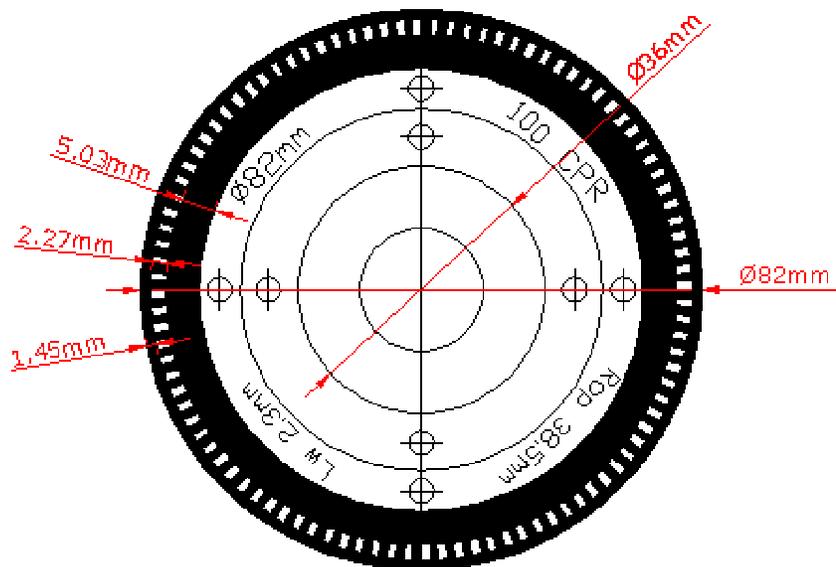


Figura 15. Disco codificador construído para o Sonic Shark.

A Figura 16 ilustra a montagem do sensor e do disco na estrutura do Sonic Shark.

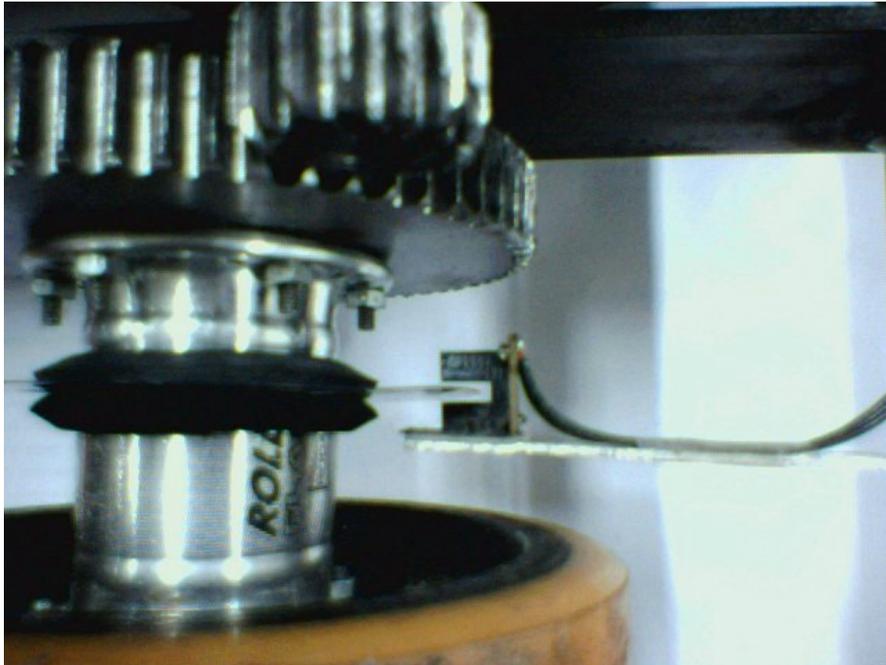


Figura 16. Foto da Instalação do Codificador Óptico.

3.3 Controle de Velocidade

Com o objetivo de se ter uma boa velocidade de resposta do sistema e erro nulo em regime permanente, optou-se por implementar um controle de velocidade baseado no controlador proporcional-integral.

Como o sistema de potência utilizado no Sonic Shark possibilita a inversão do sentido de rotação do motor, o controlador foi implementado de forma a contemplar tanto referências positivas quanto negativas de velocidade. Para saber o sentido de rotação do motor, o controlador usa a própria informação de saída do CW552 para inversão de sentido.

Como o robô possui tração diferencial, foi necessária a implementação de dois controladores PI independentes, um para cada roda. Em software o sinal de controle utilizado foi o PWM de 8 bits do hardware de controle, cujos valores variam de 0 a 255. Para obter um

sinal de controle negativo foi usado um bit em separado para a representação do sinal. Sendo assim, o sinal de PWM enviado para a planta varia sempre de 0 a 100% e o bit de inversão do motor indica se o sinal é negativo ou positivo.

Como o codificador óptico utilizado para a medição da velocidade possui apenas um canal e o sistema não possui nenhum sensor específico para a medição do sentido de rotação da roda, foi implementada em código uma condição que permite que o sinal aplicado no motor só seja negativo se a velocidade de referência for negativa. O mesmo ocorre para valores positivos, ou seja, o sinal aplicado no motor somente será positivo se a velocidade de referência for positiva.

Devido à grande dificuldade na coleta de dados para fazer o ajuste dos ganhos do controlador proporcional-integral, este foi feito de maneira heurística. Inicialmente foi feito o ajuste do ganho proporcional, de forma que o robô chegasse mais rapidamente em sua velocidade de referência sem que houvesse sobrepasso. Logo em seguida, o canal integral foi ajustado de forma a fazer com que houvesse erro nulo em regime permanente.

O ajuste do canal integral pôde ser validado da seguinte maneira: com o robô em bancada, foi escolhida uma velocidade padrão de 1 rotação por segundo da roda, ou seja, foi obtido na saída do codificador óptico um trem de pulsos de 100 pulsos por segundo, uma vez que o disco codificador possui 100 janelas por revolução. Este sinal pôde ser medido no osciloscópio e foi então observado que o valor medido ficava em torno de 100Hz, mostrando o erro nulo em regime permanente, e validando então o funcionamento do canal integral.

3.4 Implementação do Controle de Trajetória

A implementação do controle de trajetória é um dos tópicos principais deste trabalho. Este nível é vital para que seja possível a implementação de um sistema de navegação e serve para responder à pergunta “como vou?”. Seu modelo, como descrito no

sub-capítulo 2.4, permite explorar características de mobilidade do robô em trajetórias escalonadas por um sistema de navegação através do modelamento proposto.

Sua implementação consistiu na aplicação das equações (14) a (20) no software embarcado e ainda a integração deste nível com o sistema de navegação e o sistema de controle de velocidade.

O ajuste dos ganhos K_Γ e K_θ , que deveria ter sido feito através somente da escolha dos valores de Γ_c , R e v_p , acabou sendo feita de forma heurística. Isto porque, apesar do controlador proporcional de trajetória considerar a dinâmica do sistema uma perturbação, o Sonic Shark possui o seu centro de gravidade bem deslocado do ponto P , o que acentua as características não lineares do sistema e dificulta o ajuste dos ganhos do controlador. Sendo assim, este ajuste foi feito de forma a suavizar o controle de trajetória e evitar que o robô divergisse de sua trajetória.

Além disso, o controle de trajetória é realizado a cada 50ms, o que faz com que os erros de trajetórias se tornem grandes o suficiente a ponto de saturar o controlador, caso K_Γ e K_θ sejam elevados. Dessa forma, como o hardware não possibilitava a redução do período de amostragem do controle de trajetória, o procedimento heurístico tornou-se inevitável, uma vez que foi necessário um compromisso entre a não saturação do controlador, bom tempo de resposta do sistema e ainda a garantia da convergência.

Como o controlador só garante estabilidade e convergência para trajetórias compostas por segmentos de retas e curvas suaves, foi implementado um módulo de detecção de falha na trajetória. Caso o robô faça manobras com ângulos superiores a 90° , ele aborta a operação e pára imediatamente. Isto garante a segurança caso o controlador de trajetória entre numa região de instabilidade.

3.5 Implementação da Odometria

Dada as equações (23) e (24), substituindo-as nas equações de odometria (4), (5) e (6), tem-se que

$$x(k) = x(k-1) + \frac{\pi r}{N} (n_e + n_d) \cos(\phi(k)), \quad (25)$$

$$y(k) = y(k-1) + \frac{\pi r}{N} (n_e + n_d) \text{sen}(\phi(k)), \quad (26)$$

$$\phi(k) = \phi(k-1) + \frac{2\pi r}{bN} (n_d - n_e), \quad (27)$$

com n_d e n_e sendo o número de pulsos gerados pelo codificador óptico das rodas motrizes da direita e esquerda, respectivamente, através de interrupções externas, entre duas amostragens consecutivas da posição por odometria.

Sendo assim, para o cálculo da odometria no sistema de controle do Sonic Shark apenas foram implementadas em software as equações (25), (26) e (27).

3.6 Sistema de Navegação

A abordagem dada à abstração de um sistema de navegação neste trabalho segue a linha mostrada no capítulo da arquitetura de controle, conforme ilustra a Figura 13. O sistema parte do pressuposto que os níveis hierarquicamente inferiores estão funcionando de maneira adequada, uma vez que ele não é realimentado com informações provenientes destes níveis.

Este sistema engloba a odometria e o escalonador de trajetórias. O primeiro serve para tentar responder à pergunta “onde estou?” e ainda dá suporte ao controle de trajetória através do fornecimento das variáveis de trajetória, Γ e θ . Já o escalonador de trajetórias

provê um ângulo de trajetória α e uma distância a ser percorrida L , como pode ser visto na figura 17.

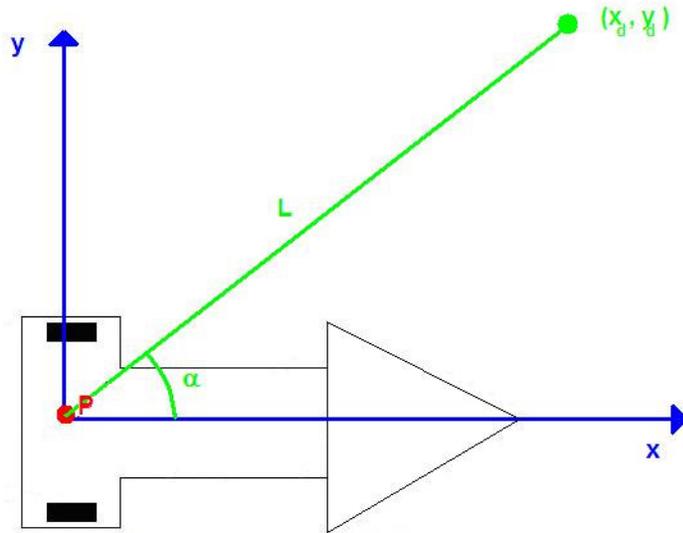


Figura 17. Coordenadas polares do ponto de destino da trajetória

Sendo assim, dado um destino (x_d, y_d) , o escalonador de trajetórias passa para o controlador de trajetórias os parâmetros α e L , dados pelas equações (28) e (29):

$$\alpha = \arctan\left(\frac{y_d}{x_d}\right), \tag{28}$$

$$L = \sqrt{x_d^2 + y_d^2}. \tag{29}$$

Para trajetórias mais genéricas, o sistema de navegação faz uma aproximação por concatenação de sucessivos segmentos de retas, de forma que a cada novo segmento, o sistema de coordenadas é zerado.

Caso haja alguma falha no controle de trajetória e este não informar a existência do erro ao sistema de navegação, o robô estará em algum ponto não previsto e o sistema de navegação não terá conhecimento desta falha.

3.7 Construção da Célula Ultrasônica

Com o objetivo de realizar um sensoriamento por ultra-som e em seguida fazer o processamento dos dados de forma a aprimorar o sistema de navegação, propõe-se um módulo sensor ultra-sônico capaz de medir distâncias de 0,2m até 4m. O objetivo é fazer com que o sistema completo tenha seis células excitadas em série que fornecerão dados de distância de objetos detectados no ambiente ao redor do robô, para que o mesmo possa escalonar trajetórias de desvio de obstáculos ou de ataque ou defesa.

A construção de cada célula leva em conta o espaço disponível para instalação do módulo. Assim sendo, ele não possui grandes dimensões (53x61mm). Seu circuito contém as seguintes partes e funções conforme os diagramas apresentados nas Figuras 18 e 19:

- a) Um gerador ajustável de pulsos elétricos de ultra-som em 40kHz baseado no astável NE555, disparado por borda de descida, ou seja, um pulso 0 lógico de controle INIT ligado ao pino RESET (pino 4) do astável;
- b) Um atuador em ponte baseado no CMOS 4069 usado para aplicar o sinal de 40kHz no transmissor;
- c) Um transdutor emissor de ultra-som, marca Murata, modelo MA40S4S;
- d) Um transdutor receptor ultra-sônico, marca Murata, modelo MA40S4R;
- e) Um amplificador operacional de baixo ruído baseado no LM833 que eleva em 60dB (1000 vezes) o sinal de eco;
- f) Um detector de pico baseado em dois diodos 1N4148. Ele detecta o invólucro positivo do sinal amplificado;
- g) Um comparador de tensão ajustável baseado no Operacional de uso geral MC1458. Este permite calibrar a margem de detecção do sinal de eco (*Threshold*);

- h) Um circuito driver de saída baseado em um BC547 para compatibilizar o sinal à arquitetura TTL do módulo de controle.

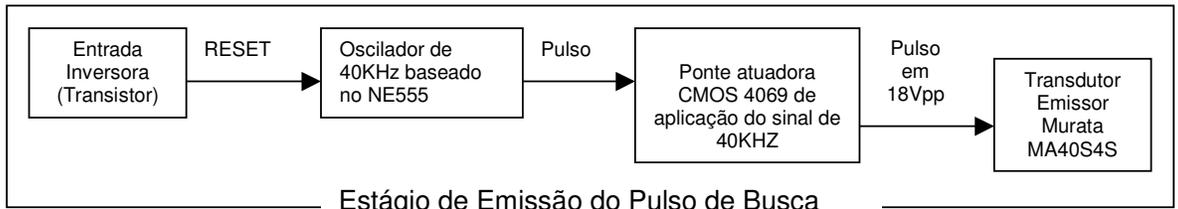


Figura 18. Diagrama de saída do Módulo Ultra-som

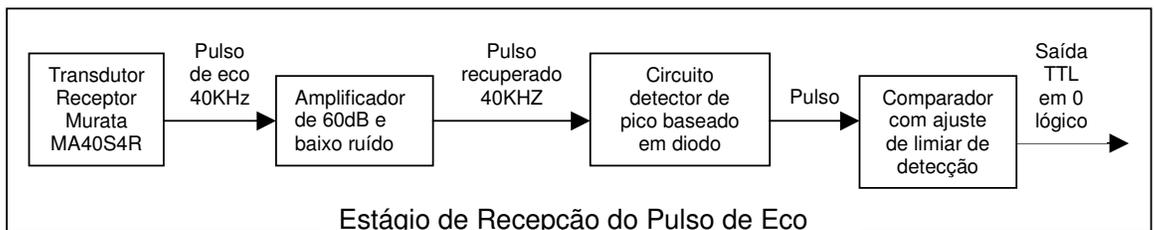


Figura 19. Diagrama de entrada do Módulo Ultra-som.

Detalhes da montagem do módulo podem ser vistos nas Figuras 21 e 22.

Discute-se a partir de agora a maneira de como tratar os intervalos referentes à medição do tempo de vôo do pulso ultra-sônico. A Figura 20 mostra as variáveis de tempo envolvidas no processo de medição com precisão viável.

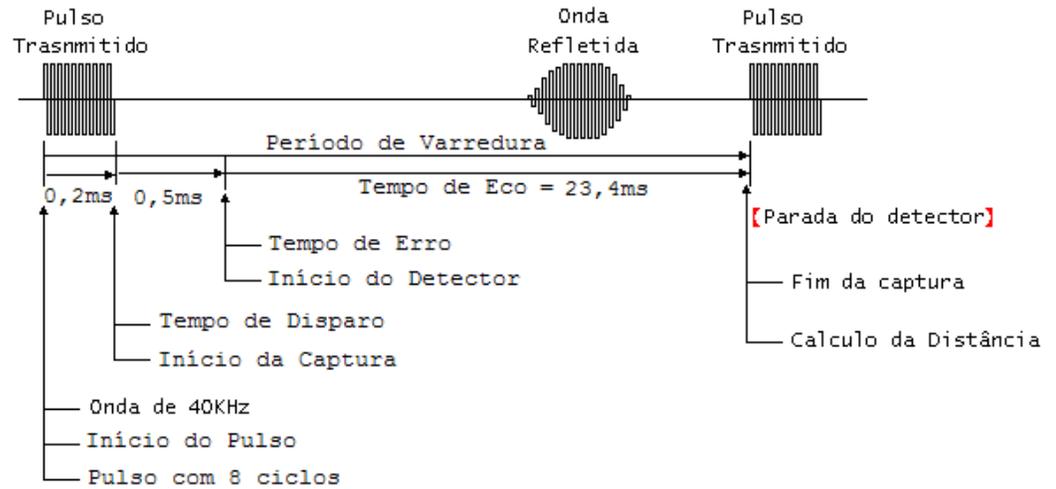


Figura 20. Tempos envolvidos na medição de distância.

O objetivo é primeiramente conseguir um pulso de eco capaz de vencer a inércia do sensor ultra-sônico e ainda sobrar energia para sobrepôr o ruído inerente ao circuito amplificador de alto ganho (60dB) e sensibilizar o detector a diodo. Para isso foi escolhido o número de 8 (oito) ciclos de ultra-som, isto é, oito vezes o período da onda de 40kHz. Calculando o **tempo de disparo**:

$$TempoDeDisparo = 8 \cdot \frac{1}{40000Hz} = 0,0002s = 200\mu s. \quad (30)$$

Esta é a largura que o pulso terá para vencer a mecânica do sensor e o ruído.

A distância mínima de 0,2m é uma especificação do transdutor Murata MA40S4S. Desta forma o **tempo de eco mínimo** deve ser:

$$TempoDeEcoMinimo = 2 \cdot \frac{0,2m}{343m/s} = 1,17ms. \quad (31)$$

Assim o tempo de eco mínimo pode conter ainda o **tempo de erro** de 0,5ms o qual tem a função de não deixar que o detector capture o próprio sinal de saída do emissor. É

uma margem de tempo para evitar esse tipo de erro de leitura. Contudo, levando em conta o tempo de eco mínimo, este mesmo já é suficiente para tal.

Para a distância máxima de 4m, o **tempo de eco máximo** fica:

$$\text{TempoDeEcoMaximo} = 2 \cdot \frac{4m}{343m/s} \cong 23,4ms . \quad (32)$$

Somando o tempo de disparo e o tempo de eco máximo tem-se que o **período de varredura** é:

$$\text{PeriodoDeVarredura} = 0,2ms + 0,5ms + 23,4ms = 24,1ms . \quad (33)$$

Para que haja uma maior precisão a distâncias pequenas, deve haver recurso computacional para uma resolução de tempo suficiente para medir décimos de milissegundo. Fazendo uma escolha através deste critério, isto é, uma interrupção computacional a cada 100 μ s, tem-se uma **resolução da medida** em metros de:

$$\text{ResoluçãoDaMedida} = 343m/s \cdot \frac{100\mu s}{2} \cong 0,01715m = 17,15mm . \quad (32)$$

Com esta resolução o sistema de navegação pode medir distâncias com erros de menos de 20mm, mais que suficiente para o que se pretende neste projeto, tendo em vista que a própria variação com a temperatura do meio inerente ao sistema acústico pode gerar erros maiores. E neste trabalho não haverá a preocupação com esta variação de modo que o tema poderá ser aprofundado em outra oportunidade.

Os tempos calculados neste estágio devem ser usados pelo programa de controle, o qual deve gerar os sinais de pulsos de busca (INIT) e contar o tempo de vôo (ToF) pela chegada do sinal de retorno (ECHO). Esta implementação deve ser feita usando um sistema de interrupção por hardware ou por software, dependendo dos recursos disponíveis no módulo controlador CW552.

Pelo fato de haver ainda poucas entradas disponíveis no hardware de controle, foi construído um módulo auxiliar de forma a multiplexar os sinais de ida e volta das seis células.

Este possui um decodificador para o pulso INIT e um multiplexador para o pulso ECHO. O primeiro usa três pinos de saída do CW552 para selecionar o sinal INIT do módulo ultra-som para medição. O segundo faz a multiplexação do sinal de ECHO através dos mesmos pinos de seleção do decodificador e enviando-o para a interrupção externa (INT0) do 80C552.

Para a construção deste módulo auxiliar (Figura 31) foi escolhido o codificador TTL 74LS138 de 3 por 8 e o multiplexador 74LS151 de 3 por 8 entradas. Através deste circuito, o controlador selecionará o módulo que irá medir a distância pelo *TempoDeDisparo* = $200\mu\text{s}$ de forma que a própria seleção envie o sinal INIT através do codificador. Estes três pinos de seleção também estão ligados ao multiplexador que deverá receber o ECHO e direcioná-lo à interrupção por hardware. Com este módulo auxiliar será possível fazer a leitura dos obstáculos em torno do Sonic em seqüência e utilizando para isso somente três pinos de saída e uma interrupção externa da placa de controle.

Em anexo encontram-se os esquemáticos do módulo ultra-sônico e do módulo auxiliar construídos. O próximo passo é a elaboração e integração em código embarcado para controle do leitor ultra-sônico de maneira que as leituras sejam utilizadas pelo sistema de navegação. As ilustrações das figuras 21 e 22 mostra o tamanho real do módulo de ultra-som construído.



Figura 21. Foto comparativa do módulo Ultra-sônico.

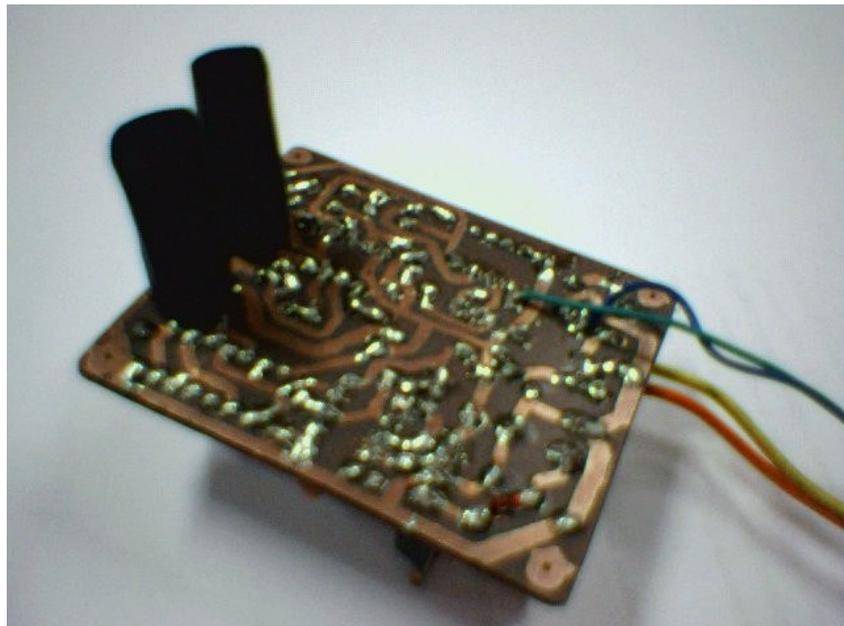


Figura 22. Foto do Módulo de Ultra-som Montado com Dutos Direcionadores.

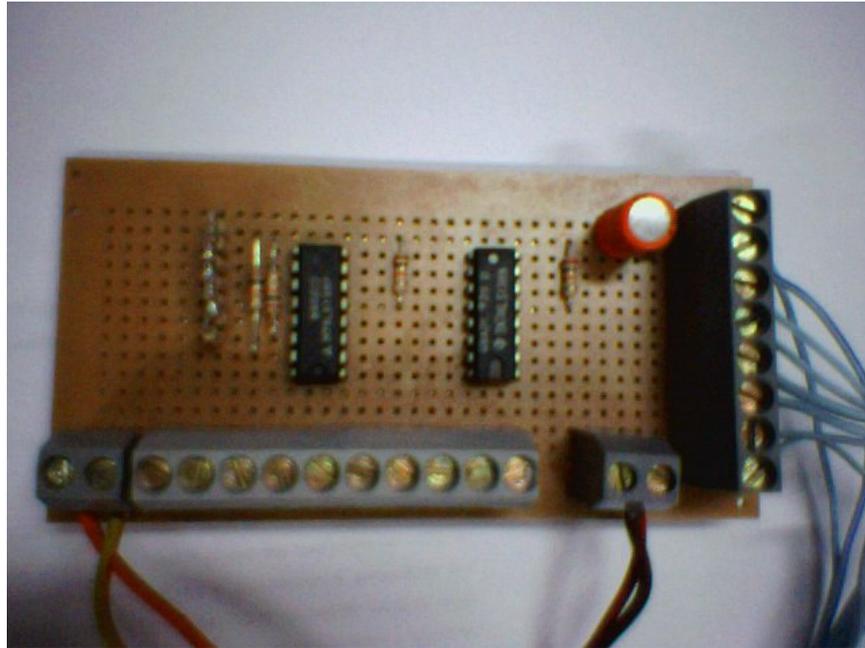


Figura 23. Foto do módulo auxiliar multiplexador.

4 Avaliação Experimental

4.1 Resultados Finais.²

4.1.1 Controle de Velocidade.

Uma das maiores dificuldades encontradas neste projeto foi em escolher os parâmetros do controlador de velocidade. Isto porque, para uma avaliação experimental rigorosa, seria necessário levantar a resposta do sistema dado os ganhos K_p e K_i . No entanto, atualmente o sistema de comunicação por rádio do Sonic Shark só está funcionando no modo *simplex*, de forma a tornar inviável o envio de dados do robô para um computador.

Sendo assim, uma alternativa seria fazer a coleta dos dados via cabo. Contudo, o controle de velocidade no Sonic Shark só trabalha de forma satisfatória para velocidades acima de 0,5 m/s, o que impossibilitou o levantamento *online* das curvas, pois é inviável manter o robô em velocidades acima disso estando preso a um cabo.

Uma outra estratégia seria guardar os valores na memória externa do hardware de controle e fazer o levantamento *offline* das curvas. Entretanto o sistema encontra-se totalmente sobrecarregado, de forma a inviabilizar este procedimento.

Devido a esses fatores, os parâmetros do controlador de velocidade foram ajustados de forma heurística e a avaliação experimental consistiu em determinar uma velocidade de referência igual para as duas rodas do robô, de forma que ele simplesmente andasse em linha reta com a velocidade pré-determinada. Sendo assim, o robô foi colocado em um corredor e uma linha reta de referência foi usada para ver o quão distante o ponto P

² Com o objetivo de mostrar os resultados de uma forma mais consistente, anexo ao trabalho encontra-se um CD (*compact disk*) com dois vídeos que mostram o desempenho do Sonic Shark ao seguir uma trajetória em forma de octógono regular com 3 metros de lado.

(centro do eixo imaginário que liga as duas rodas de tração) ficaria desta referência para uma distância percorrida de 10 metros.

Em geral, foi observado que essa distância não era maior do que 0,5m. Este resultado já poderia ser considerado satisfatório, dado os erros sistemáticos existentes no sistema devido às suas imperfeições mecânicas. No entanto, poderiam ter sido obtidos resultados melhores se fosse possível orientar o robô de forma adequada. Isto porque, dada uma pequena variação no ângulo de saída, a variação final do ponto P em relação à referência se torna mais elevada à medida que o robô vai se locomovendo ao longo da linha reta. Além disso, imperfeições no piso podem fazer com que o robô desvie da referência. Este efeito é agravado quando a velocidade do robô se torna maior, uma vez que as rodas podem derrapar.

4.1.2 Controle de Trajetória.

A avaliação experimental do controle de trajetória também não pôde ser feita de forma consistente devido aos mesmos motivos relacionados à avaliação do controle de velocidade. Entretanto, a avaliação foi feita de duas maneiras: a primeira consistiu em fazer um experimento semelhante ao do controle de velocidade, ou seja, o robô foi colocado em um corredor e uma linha reta de referência foi usada para ver o quão distante o ponto P (centro do eixo imaginário que liga as duas rodas tracionadas) ficaria desta referência para uma distância percorrida de 10 metros.

Os parâmetros que foram passados para o robô foram o ângulo ϕ (que neste primeiro caso foi de 0°) e a velocidade v_p . O robô, em geral, teve um erro de trajetória de aproximadamente 0,3m depois de percorridos os 10m. Este erro pode ter sido ocasionado pelos mesmos motivos que afetaram a avaliação do controle de velocidade, ou seja, erros sistemáticos que acabam afetando a odometria, assim como problemas com derrapagem e desnível do piso e ainda orientação incorreta. Uma maneira de minimizar o efeito da orientação incorreta seria colocar um marcador (um giz, por exemplo) e observar qual a

trajetória feita pelo robô, o que não foi feito devido ao fato de não ter sido obtidos marcadores adequados para o experimento dentro do laboratório.

A segunda maneira de se avaliar o controle de trajetória foi colocar um ângulo ϕ inicial igual a $+45^\circ$ em relação ao eixo x, conforme ilustrado genericamente na Figura 11 e uma velocidade $v_p = 1m/s$. O robô em geral fez a correção do ângulo e convergiu para a trajetória, tendo um erro Γ de aproximadamente 0,5m para $x = 10m$. O erro neste caso foi um pouco maior do que para o primeiro caso ($\phi = 0^\circ$) porque o ângulo da trajetória é corrigido abruptamente, promovendo uma derrapagem mais acentuada. O mesmo teste foi feito para um ângulo ϕ inicial igual a -45° e os resultados foram idênticos.

Avaliações mais complexas usando um escalonamento de trajetória não foram realizadas, uma vez que o ambiente de teste era restrito a um corredor relativamente estreito devido às grandes dimensões do Sonic Shark. Um outro problema do ambiente de testes foi o piso escorregadio, que ocasionou em derrapagens indesejadas e portanto um erro maior para a trajetória.

Uma maneira de se minimizar o efeito da derrapagem seria colocar um filtro discreto de primeira ordem na entrada do sistema de controle de velocidade, de forma a suavizar o sinal aplicado nos atuadores.

4.1.3 Célula Ultra-Sônica

Devido ao esgotamento dos recursos computacionais do microcontrolador utilizado e ainda à falta de tempo hábil para a elaboração e implementação de um novo hardware de controle, não foi possível fazer uma avaliação experimental consistente da célula ultra-sônica.

5 Conclusões

O Sonic Shark desde o início se mostrou um grande desafio de engenharia, principalmente devido ao seu grande porte e às suas finalidades. Desafio maior ainda foi implementar um controle de velocidade e um controle de trajetória que pudessem funcionar de maneira satisfatória e melhorar o desempenho do robô em ambientes hostis, tal como a arena de Guerra de Robôs.

O controle proporcional-integral de velocidade por si só já possibilitou manter o robô em linha reta, algo antes muito difícil de fazer usando apenas o controle em malha aberta. Já o controle de trajetória se mostrou suficiente para manter o robô dentro de uma trajetória pré-definida, possibilitando enviar o robô de uma forma mais eficiente para qualquer ponto definido a partir de sua localização atual, sendo o erro dado basicamente pela odometria.

Como o microcontrolador utilizado no projeto tem um poder computacional muito pequeno, foi extremamente difícil escalonar as tarefas de forma a manter os tempos de amostragem o mais determinísticos possível, de forma a aumentar o desempenho da arquitetura de controle. Devido ao esgotamento dos recursos internos do microcontrolador, não foi possível fazer a integração dos módulos de ultra-som com o restante do sistema. Sendo assim, o sensoriamento por ultra-som atualmente não agrega nenhum valor ao sistema de navegação.

Além disso, a dinâmica do Sonic Shark tornou-se um grande empecilho para a implementação do controle de trajetória, uma vez que a sua parte anterior possui uma grande concentração de massa, aumentando o momento de inércia do robô. Sendo assim, o controlador de trajetória fica mais sujeito às perturbações dinâmicas, o que dificultou no ajuste dos parâmetros do controlador proporcional de trajetória.

Apesar de todas as dificuldades de implementação do sistema e de coleta de dados para a validação dos resultados, pode-se considerar que a arquitetura proposta mostrou uma

performance melhor para a plataforma Sonic Shark, quando comparada com o sistema anterior. Além disso, esta plataforma se mostrou excelente para a implementação de novas técnicas de controle e sensoriamento.

Proposta para Trabalhos Futuros

Pelo fato da plataforma Sonic Shark ter se mostrado satisfatória para a aplicação de técnicas de controle, são sugeridos os seguintes trabalhos:

- desenvolvimento de um novo hardware de controle baseado em processadores mais rápidos, como o Dallas DS89C450;
- implementação de um controlador de trajetórias mais robusto, tal como o controlador não-linear (Borges et al. 2000);
- implementação de um sistema de navegação mais aprimorado, usando abordagens de mapeamento e cartografia;
- desenvolvimento de um sistema de sensoriamento mais completo, usando acelerômetros e tacômetros, fazendo então uma abordagem de fusão sensorial.
- implementação de um sistema de detecção de falhas e telemetria, juntamente com uma interface amigável implementada em PC.
- desenvolvimento de um enlace de rádio semi-duplex com um protocolo com detecção de erros.

Referências

- Adorno, B., Baptista, R. (2004). Estudo sobre o sensor de aceleração ADXL 202E e suas aplicações, *Seminário apresentado na disciplina Instrumentação de Controle na Universidade de Brasília*.
- Bellardi, T., Sousa, G. (2003). Implementação de um Sistema de Posicionamento Baseado em Referências Absolutas e Relativas para uma Plataforma Robótica Móvel Teleoperada, *Relatório de Graduação II*, Universidade de Brasília.
- Borenstein, J., Everett, H. R., Feng, L. (1996). "Where am I?" - *Systems and Methods for Mobile Robot Positioning*. Editado e compilado por J. Borenstein.
- Borges, G., Lima, A., Deep, G. (2000). Design of an Output Feedback Trajectory Controller for an Automated Guided Vehicle, *XIII Congresso Brasileiro de Automática*.
- Borges, G., Lima, A., Deep, G. (2003) Controladores Cinemáticos de Trajetória Para Robôs Móveis com Tração Diferencial, *VI Simpósio Brasileiro de Automação Inteligente*.
- Briggs, G., Moir, J. (1963) *Audio and Acoustics*, Bradford: Wharfedale Wireless Works.
- Doebelin, E. (1983). *Measurements Systems, Application and Design*. McGraw-Hill.
- Franklin, G., Powell, J., Workman, M. (1998). *Digital Control of Dynamic Systems*, 3ª Edição, Addison Wesley.
- Sedra, A., Smith, K. (2000). *Microeletrônica*, 4ª Edição, Makron Books.
- When, H., Belanger, P. (1997). Ultrasound-Based Robot Position Estimation, *IEEE Transactions on Robotics and Automation* **13** (6): 837-842.

Informações de Produtos Citados de Empresas Comerciais

MURATA – Murata Erie North America, 2200 Lake Park Drive, Smyrna, GA 30080, 800-831-9172.

6.1 Listagem Comentada do Código de Controle.

O primeiro trecho de código apresentado define as constantes e variáveis globais do sistema:

```

/*****
Universidade de Brasília - Faculdade de Tecnologia
Engenharia Mecatrônica
Trabalho de Graduação - "Sistema de Controle e Navegação do robô Sonic Shark"
Professor: Geovany Araujo Borges
Alunos: Bruno Vilhena Adorno
Sérgio Amorim Reis

ATRIBUIÇÕES DAS PORTAS
PORTA P1
P1.0 a P1.3 -> disponível para encoder (CMT0 a CMT3)
P1.4 -> VAGO
P1.5 -> BUZZER (buzina com nível 1)
P1.6 -> RX/TX = 0 TRANSMITE / 1 RECEBE (somente para o rádio half-duplex)
P1.7 -> CD = Carrier Detect, sinal binário de presença de portadora no rádio
=> 0=sem portadora
PORTA P4
P4.0 -> COMANDO AUXILIAR 0 (driver coletor aberto 1A) aciona com 1 LED VERDE
P4.1 -> COMANDO AUXILIAR 1 (driver coletor aberto 1A) aciona com 1 LED VERDE
P4.2 -> COMANDO AUXILIAR 2 (driver coletor aberto 1A) aciona com 1 LED VERDE
P4.3 -> COMANDO AUXILIAR 3 (driver coletor aberto 1A) aciona com 1 LED VERDE
P4.4 -> dir.esq ???
P4.5 -> hab.esq ???
P4.6 -> dir.dir ???
P4.7 -> hab.dir ???
INT0, INT1 -> encoder
PWM0 -> Sinal de PWM para o motor esquerdo
PWM1 -> Sinal de PWM para o motor direito
PWM min = 0, PWM max = 255;
*****/
#include "80552.h" // biblioteca com a declaração dos registradores para o 80552
#include <math.h>

#define baud      0xF4
#define TH 0xFF & ((0xFFFF-((11059200/1000)/12)) >> 8)
#define TL 0xFF & (0xFFFF-((11059200/1000)/12))
// controle de velocidade
#define Tc 0.05
#define Kp 2.5 //k= ganho proporcional = 5
#define Ki 0.25
//odometria
#define N 100
#define r 0.0625
#define b 0.495
#define PI 3.1416
#define CONSTANTE_PIR_N (PI*r)/N
#define CONSTANTE_PIR_Nb (CONSTANTE_PIR_N/b)
#define CONSTANTE_2PIr_Nb (2*CONSTANTE_PIR_Nb)
#define CONSTANTE_2b_r (2*b)/r
#define CONSTANTE_100_360graus 28
#define CONSTANTE_100_2PIr 100/(2*PI*r)
#define SENTIDO_ANTI_HORARIO 1
#define SENTIDO_HORARIO 0

```

```

#define LIGADO 0
#define DESLIGADO 1
#define CHAVE1 '*'
#define CHAVE2 '-'
#define CHAVE3 '.'
#define comando_robo_direita '5'
#define comando_robo_esquerda '6'
#define comando_robo_para 't'
#define comando_velocidade_1 '<'
#define comando_velocidade_2 'G'
#define comando_velocidade_3 'K'
// *****
// **                               VARIÁVEIS GLOBAIS                               **
// *****
//Variável para uso na janela de correção e de odometria
unsigned int iMilisegundo = 0, iMilisegundoOdometria=0, iMilisegundoControleTrajetoria = 0,
tempoSetup = 0;
unsigned int habilitaOdometria = 0, habilitaControleTrajetoria = 0, iMilisegundoTeste = 0,
tempoTimeout = 0;
int habilitaDebug = 0;
// variáveis que contém os erros de medição de velocidade
float erroVelocidadeDcESQ = 0, erroVelocidadeDcESQIntegral = 0;
float erroVelocidadeDcDIR = 0, erroVelocidadeDcDIRIntegral = 0;
// Sinais de saída
float u_ESQ = 0;
float u_DIR = 0;
//variáveis para referência de tempo para os motores
unsigned int milisegDcESQ=0, milisegDcDIR=0;
// velocidade de referência e velocidade medida (pulsos por segundo)
float velocidadeReferenciaDcESQ = 0 , velocidadeMedidaDcESQ = 0;
float velocidadeReferenciaDcDIR = 0 , velocidadeMedidaDcDIR = 0;
float velocidadeReferenciaDcDIRfloat = 0, velocidadeReferenciaDcESQfloat = 0;
float KGama=0, KTeta=0;
float x = 0, y = 0;
float velocidadeRobo = 0.5;
float phi = 0, alpha = 0;
int pulsosDir = 0, pulsosEsq = 0;

```

O próximo trecho de código apresentado é a função *clock()*. Ele é responsável pelo controle de velocidade e relógio do sistema. Nele são implementados os dois controladores proporcional-integral independentes e ainda é habilitada a odometria e o controle de trajetória. A função *clock()* é uma rotina de interrupção que ocorre a cada 1ms, gerando a base de tempo para o sistema. É importante ressaltar que ela implementa somente o nível 0 da arquitetura de controle.

```

void clock(void) interrupt 1 using 2
{
    TH0=TH;
    TL0=TL;
    milisegDcESQ++; //temporizador para calcular a velocidade do motor 0
    milisegDcDIR++;

    if (iMilisegundoOdometria++==49)
    {
        habilitaOdometria = 1;
        iMilisegundoOdometria=0;
    }

    //Controlador Proporcional-Integral para a roda esquerda
    if (iMilisegundo++==19)
    {
        tempoSetup++;
        if(milisegDcESQ>=25)

```

```

    {
        velocidadeMedidaDcESQ=0;
        milisegDcESQ = 0;
    }

    erroVelocidadeDcESQ = velocidadeReferenciaDcESQ - velocidadeMedidaDcESQ;
    erroVelocidadeDcESQintegral += (erroVelocidadeDcESQ);
    if(erroVelocidadeDcESQintegral>5000) erroVelocidadeDcESQintegral = 5000;
    if(erroVelocidadeDcESQintegral<-5000) erroVelocidadeDcESQintegral = -5000;
    u_ESQ = (Kp*erroVelocidadeDcESQ)+(erroVelocidadeDcESQintegral*Ki);
    if(velocidadeReferenciaDcESQ < 0)
    {
        P4_4=SENTIDO_ANTI_HORARIO;
        if(u_ESQ < -255) u_ESQ = -255;
        if(u_ESQ > 0) u_ESQ = 0;
        u_ESQ = -u_ESQ;
    }
    else
    {
        P4_4=SENTIDO_HORARIO;
        if(u_ESQ > 255) u_ESQ = 255;
        if(u_ESQ < 0) u_ESQ = 0;
    }

//Controlador Proporcional-Integral para a roda direita

    if(milisegDcDIR>=25)
    {
        velocidadeMedidaDcDIR=0;
        milisegDcDIR = 0;
    }
    erroVelocidadeDcDIR = velocidadeReferenciaDcDIR - velocidadeMedidaDcDIR;
    erroVelocidadeDcDIRintegral += (erroVelocidadeDcDIR);
    if(erroVelocidadeDcDIRintegral>5000) erroVelocidadeDcDIRintegral = 5000;
    if(erroVelocidadeDcDIRintegral<-5000) erroVelocidadeDcDIRintegral = -5000;
    u_DIR = (Kp*erroVelocidadeDcDIR)+(erroVelocidadeDcDIRintegral*Ki);
    if(velocidadeReferenciaDcDIR < 0)
    {
        P4_6=SENTIDO_HORARIO;
        if(u_DIR < -255) u_DIR = -255;
        if(u_DIR > 0) u_DIR = 0;
        u_DIR = -u_DIR;
    }
    else
    {
        P4_6=SENTIDO_ANTI_HORARIO;
        if(u_DIR > 255) u_DIR = 255;
        if(u_DIR < 0) u_DIR = 0;
    }

    PWM1 = u_DIR;
    PWM0 = u_ESQ;
    iMilisegundo = 0;
}

}

```

A medição de velocidade é feita através de interrupções externas, conforme pode ser visto nas funções de interrupção *externo0()* e *externo1()*:

```

void externo1(void) interrupt 0 using 1 // INT0
{
    if(velocidadeReferenciaDcESQ >= 0) pulsosEsq++;
    else pulsosEsq--;
    if(milisegDcESQ)
    {
        if (P4_4==SENTIDO_ANTI_HORARIO){
            velocidadeMedidaDcESQ = -(int)(1000/milisegDcESQ);
        }
        else{
            velocidadeMedidaDcESQ = (int)(1000/milisegDcESQ);
        }
    }
}

```

```

    }
  }
  milisegDcESQ=0;
}
void externo0 (void) interrupt 2 using 1 // INT1
{
  if(velocidadeReferenciaDcDIR >= 0) pulsosDir++;
  else pulsosDir--;
  if(milisegDcDIR)
  {
    if (P4_6==SENTIDO_HORARIO){
      velocidadeMedidaDcDIR = -(int)(1000/milisegDcDIR);
    }
    else{
      velocidadeMedidaDcDIR=(int)(1000/milisegDcDIR); //frequência do sinal
    }
  }
  milisegDcDIR=0;
}

```

A odometria e o controle de trajetória são implementados dentro da função principal e estão dentro de um *loop* infinito, conforme podemos ver através da função *main()*. Nesta função é implementada também a detecção de falha na trajetória.

```

void main(void)
{
  long int pulsosDirAux = 0, pulsosEsqAux = 0;
  float delta;
  long int iInicioTrajetoria = 1, iExecutandoTrajetoria = 0;

  float raioTrajetoria = 0.3, gamaC = 0.2;
  float deltaOmega=0, teta=0, gama=0;
  float L = 0;

  setRegistros();

  P4_7=DESLIGADO;
  P4_5=DESLIGADO;
  P1_0 = 1;
  P1_1 = 1;
  while(tempoSetup < 1500)
  {
    if(tempoSetup < 50) P1_5 = 1;
    if(tempoSetup >= 50) P1_5 = 0;
    if(tempoSetup >=1300) P1_5 =1;
  }

  P1_5 = 0;
  pulsosDir = pulsosEsq = 0;
  errovelocidadeDcESQ = errovelocidadeDcESQintegral = 0;
  errovelocidadeDcDIR = errovelocidadeDcDIRintegral = 0;
  u_ESQ = 0;
  u_DIR = 0;
  milisegDcESQ = milisegDcDIR=0;
  velocidadeMedidaDcESQ = 0;
  velocidadeMedidaDcDIR = 0;
  velocidadeReferenciaDcDIRFloat = velocidadeReferenciaDcESQFloat = 0;
  velocidadeReferenciaDcESQ = 0;
  velocidadeReferenciaDcDIR = 0;

  P4_7=LIGADO; //Habilita motor direito
  P4_5=LIGADO; //Habilita motor esquerdo
  P4_6=SENTIDO_ANTI_HORARIO;
  P4_4=SENTIDO_HORARIO;
}

```

```

x = 0;
y = 0;
phi = 0;

KTeta = 15;
KGama = 0.9;

while(1)
{
    if (habilitaOdometria==1)
    {
        habilitaOdometria=0;
        pulsosDirAux = pulsosDir;
        pulsosEsqAux = pulsosEsq;
        pulsosDir = 0;
        pulsosEsq = 0;

        delta = CONSTANTE_PIR_N * (pulsosEsqAux + pulsosDirAux);
        x += delta * cos(phi);
        y += delta * sin(phi);
        phi += CONSTANTE_PIR_Nb * (pulsosDirAux-pulsosEsqAux);

        teta= - phi;
        gama = y/cos(phi);
        deltaOmega = (KGama*gama)+(KTeta*teta);

        velocidadeReferenciaDCDIR = (velocidadeRobo + (r*deltaOmega))*CONSTANTE_100_2PIr;
        velocidadeReferenciaDCESQ = (velocidadeRobo - (r*deltaOmega))*CONSTANTE_100_2PIr;

        if((phi>(PI/2))|| (phi<(-PI/2)))
        {
            velocidadeRobo = 0;
            KGama = 0;
            KTeta = 0;
        }
    }
}
}

```

A função *setRegistros()* carrega os registradores do microcontrolador com os dados necessários para o correto funcionamento do *timer*, frequência do PWM, comunicação serial e das interrupções externas, conforme vemos a seguir:

```

void setRegistros(void)
{
    IEN0=0;           ///  

    IP0=0;            ///  

    TCON=0;           ///  

    TMOD=0;           ///  

    SOCON=0;          ///  

    PCON=0;           ///  

    /* Registrador IE */  

    ET0=1;            ///  

    EX0=1;            ///  

    EX1=1;            ///  

    /* Registrador TCON */  

    IE0=0;            ///  

    IE1=0;            ///  

    IT0=1;            ///  

    IT1=1;            ///  

    TR1=1;            ///  

    TR0=1;            ///  

    /* Registrador IP */  

    PT0=0;            ///  

}

```

```

/* Registrador TMOD */
// timer1 - setado para 8 bit auto-reload timer/counter xx10 xxxx
// timer0 - 16 bits timer counter xxxx xx01
TMOD=0x21; // 0010 0001
/* Registrador S0CON */
S0CON=0x52; // 0101 0010
RI=0; // Modo 01 de Transmissao; REN=1 e TI=1;
TH1=baud;
P1=0x00;
P4=0x00;
PWM0=0;
PWM1=0;
PWMP=0x6f;
EA=1;
}

```

6.2 Plataforma Sonic Shark

O robô foi construído visando o ingresso da universidade no evento Guerra de Robôs do Encontro Nacional de Estudantes de Engenharia de Controle e Automação – ENECA. Seu projeto foi dividido em três partes para que fossem integradas em um só equipamento. São elas:

6.2.1 Estrutura Mecânica

A plataforma Sonic Shark possui boa resistência mecânica, o que contribui para torná-lo relativamente pesado e ainda haver uma concentração irregular de massa, o que provê ao protótipo uma dinâmica singular.

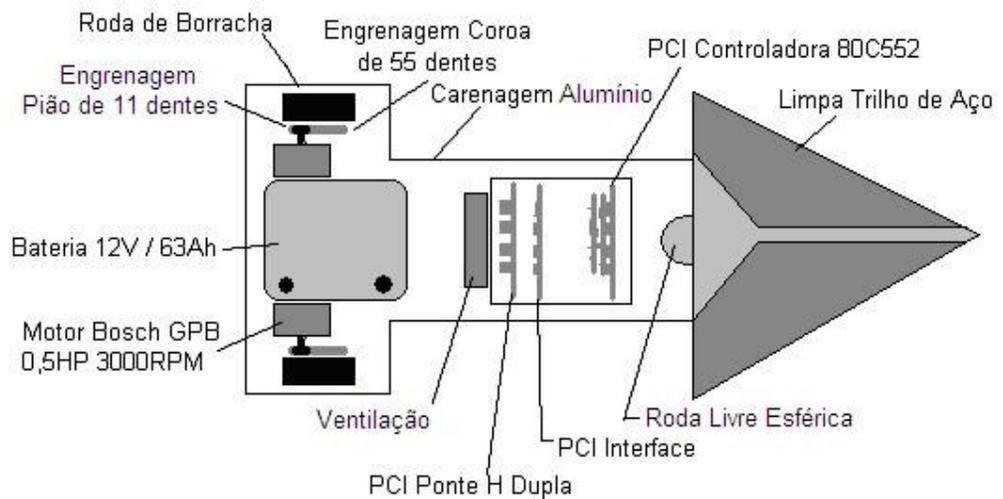


Figura 24. Visão superior da estrutura mecânica e seus componentes.

Sua construção foi feita sobre um chassi preexistente de um brinquedo e este foi reforçado e remodelado para suportar a idéia preconcebida de um formato de flecha. Sua carenagem de alumínio não concentra grande quantidade de massa para que o peso final não exceda o limite de 50 kg definido pela regra do evento.

- a) Peso: 45Kg
- b) Comprimento: 1,10m
- c) Largura máxima: 0,75m

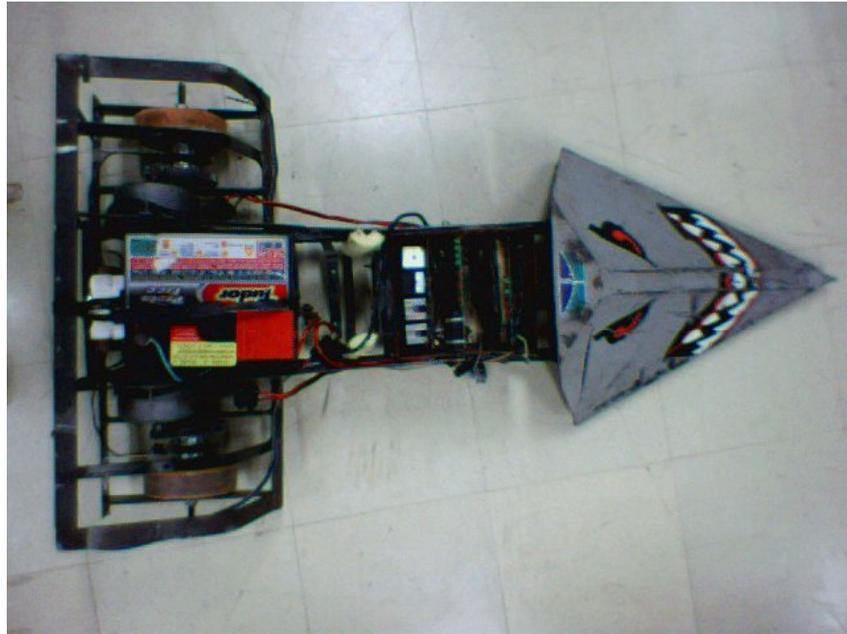


Figura 25. Foto real da estrutura mecânica do Sonic.

6.2.2 Projeto Eletro-eletrônico

Desempenha as funções elementares de receber dados remotos através de um protocolo particular, sensoriar elementos locais gerando dados processáveis, processar dados locais e remotos e acionar os atuadores locais. Estas tarefas fundamentam o sistema eletrônico do Sonic Shark. Para isso, este foi dividido em estruturas modulares que são conectadas via PCI de base.

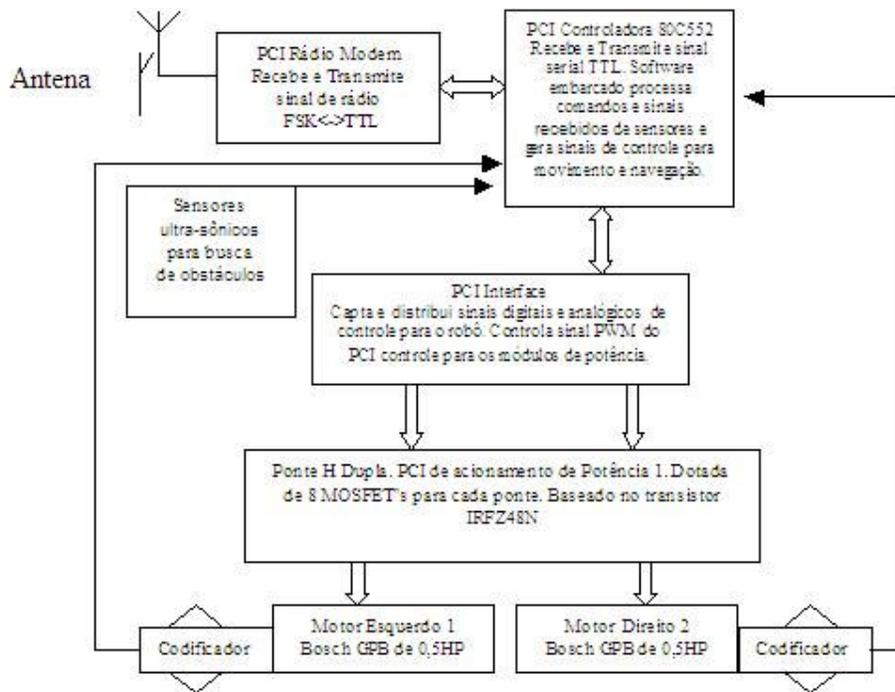


Figura 26. Diagrama modular do sistema eletrônico do Sonic Shark.



Figura 27. Foto do Sistema Eletro-eletrônico.

6.2.2.1 PCI Controladora

Criada pela Empresa *Controlware* (Kit didático CW552), é baseada no microcontrolador Phillips 80C552, que tem como vantagem a implementação já embarcada de PWM. A placa recebe o sinal via RS232 e o programa de controle executa os comandos conforme protocolo previamente definido. A saída (I/O), com os sinais de PWM e comandos (Bit 0 e 5V), são enviadas para as PCI's de potência (ponte H) via cabos coaxiais e conectores do tipo porta-clip. Este módulo é todo alimentado por 5V regulado, mas não filtrado, dos 12V da bateria pelo CI 7805 instalado na própria placa.

6.2.2.2 Módulo Rádio Controle

Atualmente usa um receptor FM UHF de dados codificados em FSK, modelo Radiometrix BIM-433-10 de 433,92MHz e possui capacidade de transmissão de dados de até 10Kbps. O sinal de rádio frequência (RF), proveniente do transceptor ligado à HP48G, chega por uma antena camuflada na carenagem do Sonic Shark, feita com o próprio cabo, o qual é decodificado para um sinal TTL (0 e 5V), segue para uma interface RS232 (+12V e -12V) e em seguida é enviado para a placa de controle por um cabo blindado (coaxial). Este módulo é todo alimentado por 5V regulado, mas não filtrado, dos 12V da bateria pelo CI 7805 instalado na sua própria placa.

6.2.2.3 Módulo Interface

Esta placa possui várias funções conforme relacionadas a seguir:

a) Acionar com proteção a Ponte H:

Esta função é implementada por lógica combinacional e é extremamente necessária para evitar, sob qualquer circunstância, o acionamento simultâneo de dois ou mais transistores em paralelo à bateria. Isto colocaria os terminais em curto-circuito e os MOSFETs não suportariam as altas correntes.

b) Sinalizar através de LEDs se as fontes de tensão envolvidas no sistema estão funcionando ou estão abaixo do ideal:

Possui três LEDs que monitoram as tensões de 5V, 12V e 20V, que alimentam os circuitos TTL, acionadores em coletor aberto e a excitação do *gate* dos transistores, respectivamente.

c) Isolar o sistema de potência do sistema digital através de acopladores ópticos:

São circuitos integrados que isolam eletricamente o sistema de potência, isto é, o *gate* do MOSFET e a saída da lógica TTL. Esta isolamento é parcial uma vez que a referência de terra negativo é a mesma para os dois módulos.

d) Acionar quatro dispositivos quaisquer via quatro saídas em coletor aberto de até 1A;

São quatro transistores polarizados em coletor aberto e excitados por sinais TTL da controladora através de acopladores ópticos. Acionam cargas de até 1A.

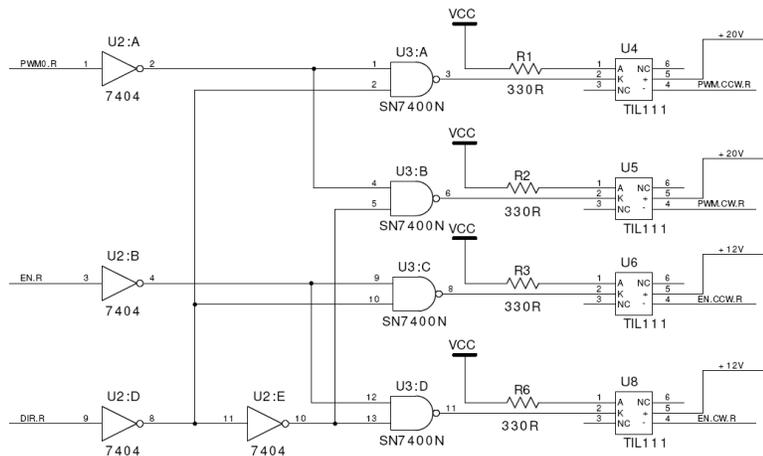


Figura 28. Lógica de proteção e isolação óptica.

6.2.2.4 Módulo de Potência – Ponte H Dupla

Cada ponte utiliza 8 transistores MOSFET canal N IRFZ48N de 64A nominal cada, para acionamento em dois sentidos dos motores do robô. Cada transistor é fixado na PCI e suporta o seu próprio dissipador de calor modelo HS2816-40 (28x16x40mm). Somando-se a capacidade nominal de cada transistor pode-se estimar que a Ponte H completa é capaz de fornecer 128A contínuo e suportar 220A de pico. Este módulo é alimentado por 12V direto da bateria para dreno de alta corrente e possui proteção de *gate* via diodo *zener* de 20V, pois é a tensão máxima suportada pelo mesmo.

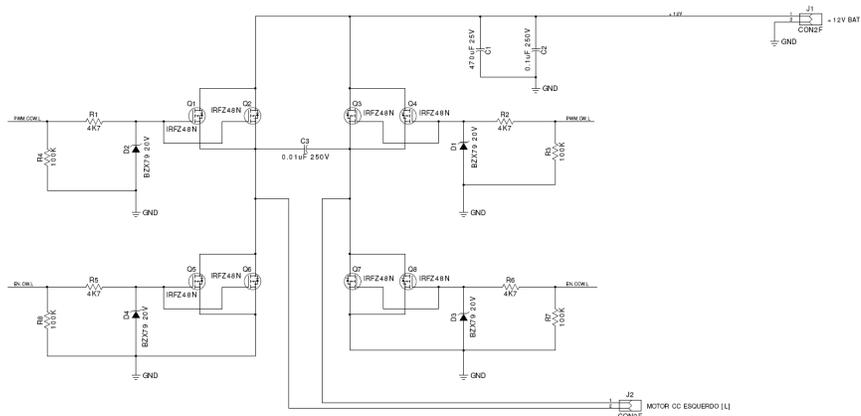


Figura 29. Circuito da ponte H.

6.2.2.5 Motor Bosch GPB 12V DC

Cada motor pode fornecer até 0,5HP (372,85W) de potência mecânica com eficiência de 74%, conforme a curva de desempenho abaixo.

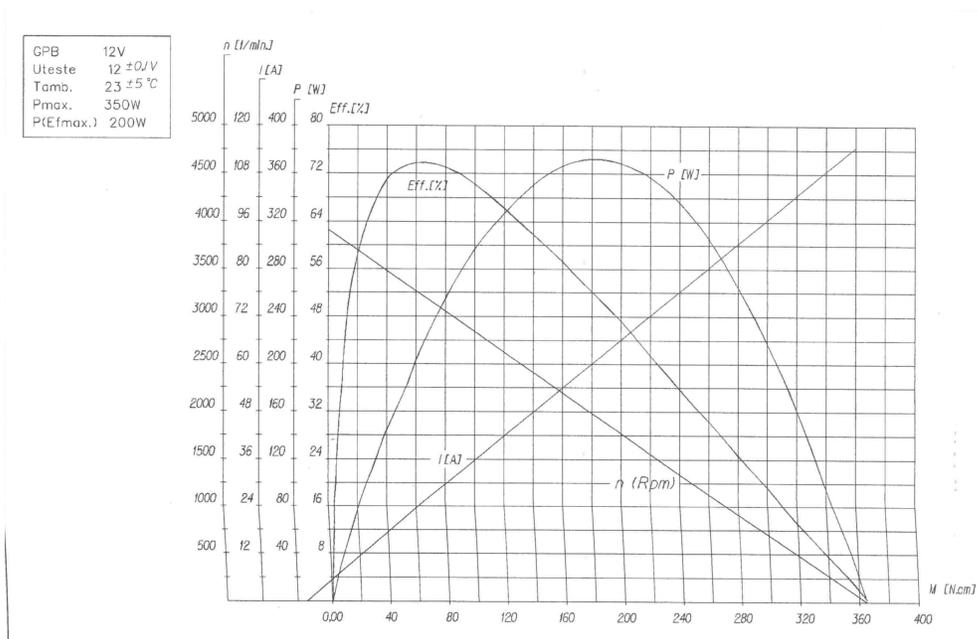


Figura 30. Curvas de desempenho do motor Bosch GPB do Sonic Shark.

6.2.3 Embarque de Controle Remoto

A conseqüente melhoria das características eletrônicas de todo o robô e a implementação do rádio interligado à controladora possibilitou um programa de controle capaz de utilizar todos os recursos eletrônicos e de sensoriamento, implementados no sistema modular proposto.

Este programa tem como principal característica o fato de ser modular e dotado de sistema de autodiagnóstico visual (*display*) e sonoro (*buzzer*), podendo ser também elaborado um protótipo mais completo, possivelmente dotado de um sistema de telemetria via rádio bidirecional.

Estas características também exigem maior estabilidade e segurança na comunicação via rádio. Para isso deve ser implementado em uma fase posterior um bom algoritmo de detecção e correção de erros de comunicação (sugere-se o CRC16). Sob estas novas exigências de controle, já está sendo usado um rádio bidirecional de maior potência, melhor sensibilidade e maior taxa (bps).

O controle remoto, atualmente a calculadora HP48, pode ser preservado desde que consiga rodar com maior velocidade o programa gerador de comandos com controle de erros. Isto pode ser conseguido migrando da linguagem atual RPL para a linguagem C padrão, sendo que um compilador cruzado (cross compiler -> XGCC) usado como aplicativo externo em um PC, compila diretamente para a linguagem de máquina da HP48.

6.3 Esquemáticos dos Circuitos Eletrônicos

6.3.1 Célula Ultra-sônica

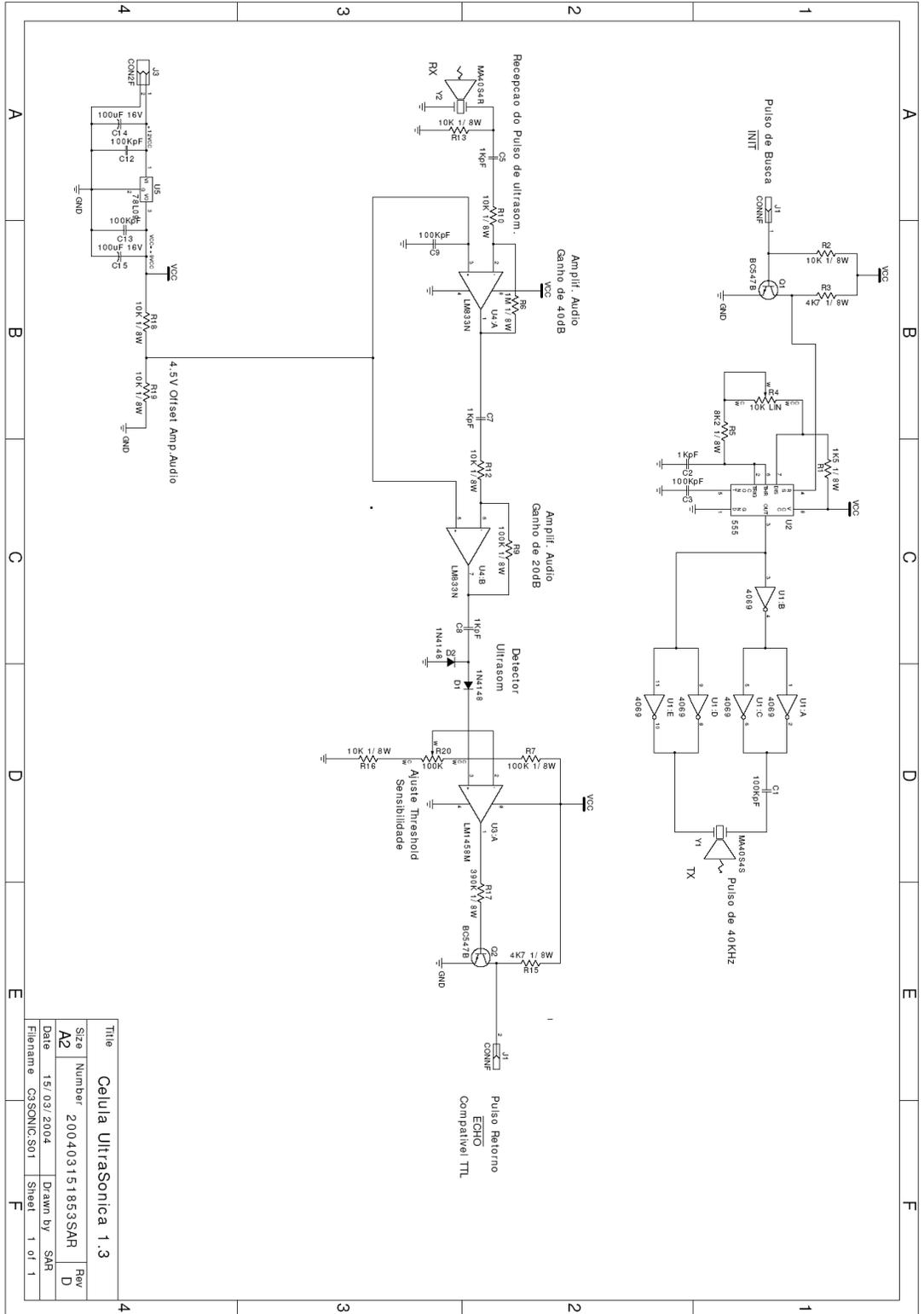


Figura 31. Esquemático Módulo Ultra-sônico.

6.3.2 Módulo Auxiliar Multiplexador (CODEC)

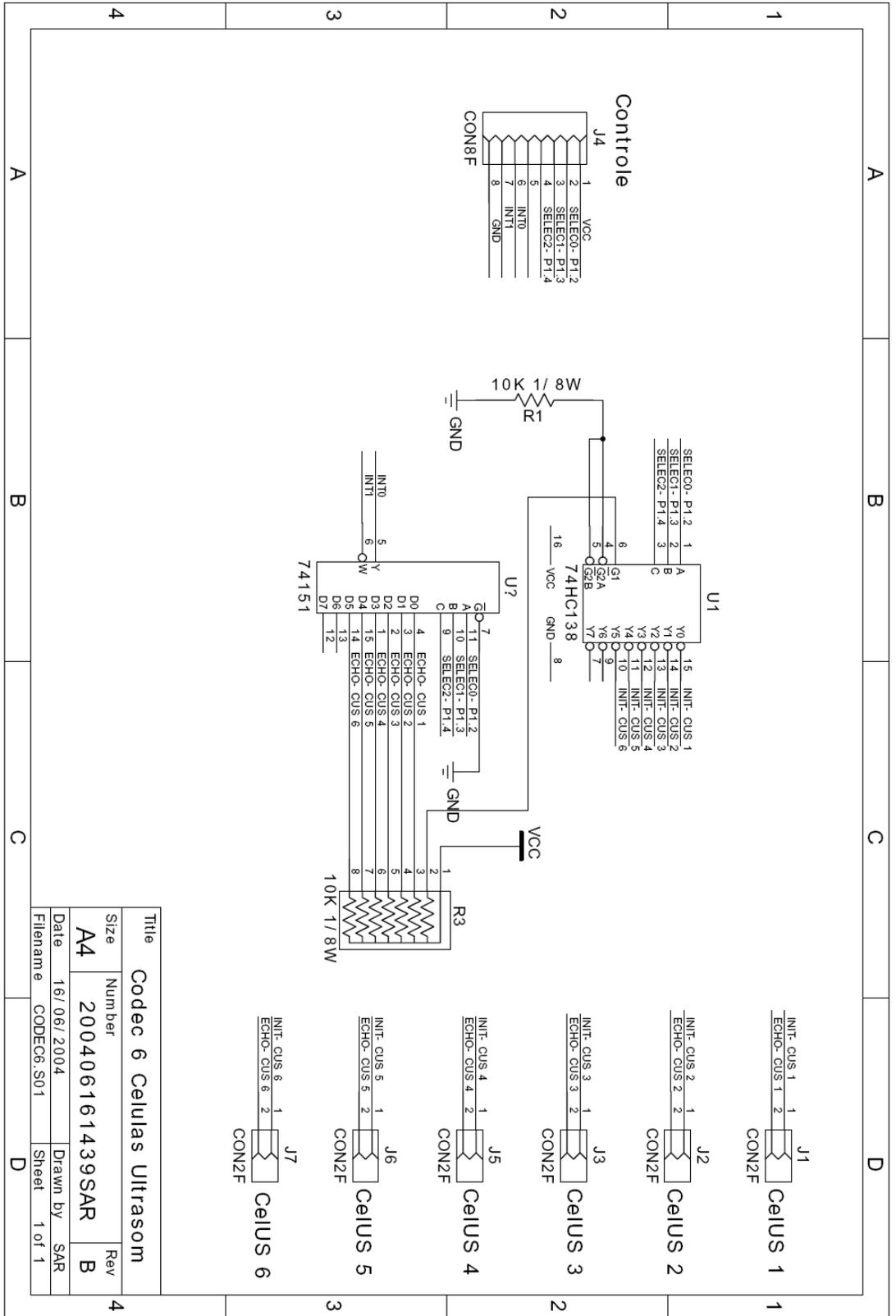


Figura 32. Esquemático Módulo Multiplexador.

6.3.3 Condicionador do Codificador Incremental

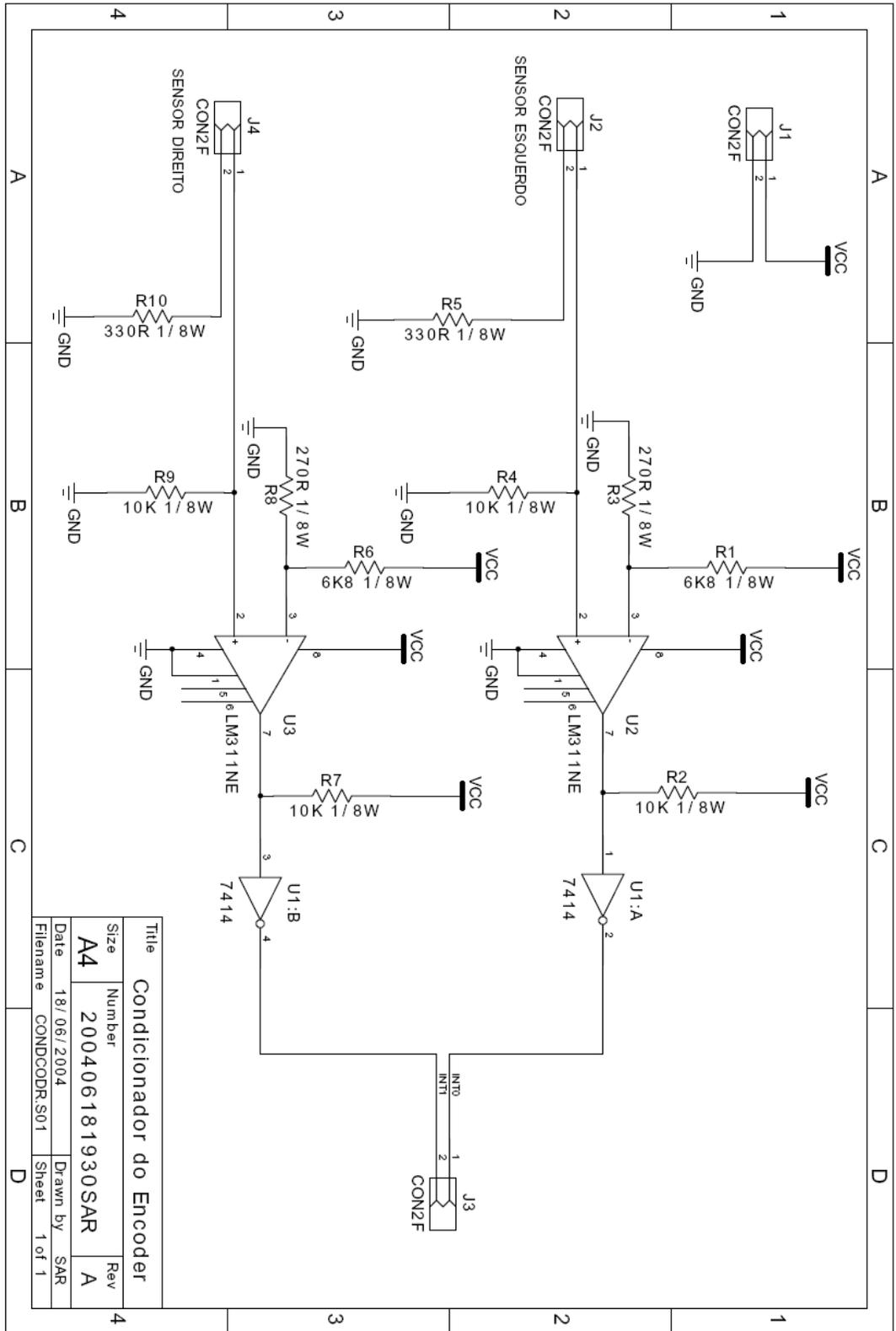


Figura 33. Esquemático Condicionador do Encoder.

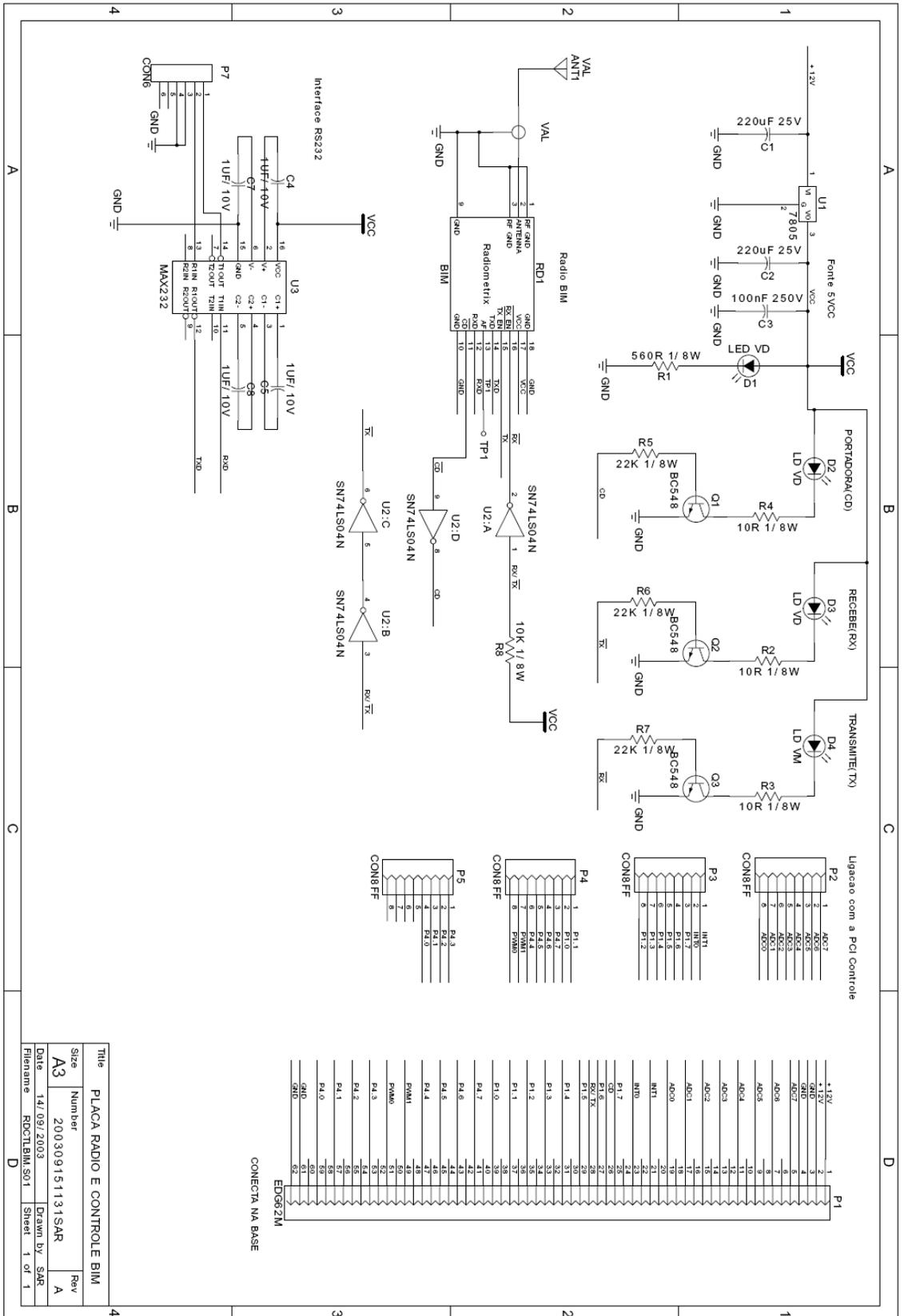


Figura 35. Esquemático PCI Base do CW552 e Rádio.

Title	PLACA RADIO E CONTROLE BIM	
Size	Number	Rev
A3	200309151131SAR	A
Date	14/09/2003	Drawn By
Filename	ROCTLBIM.S01	Sheet
		1 of 1

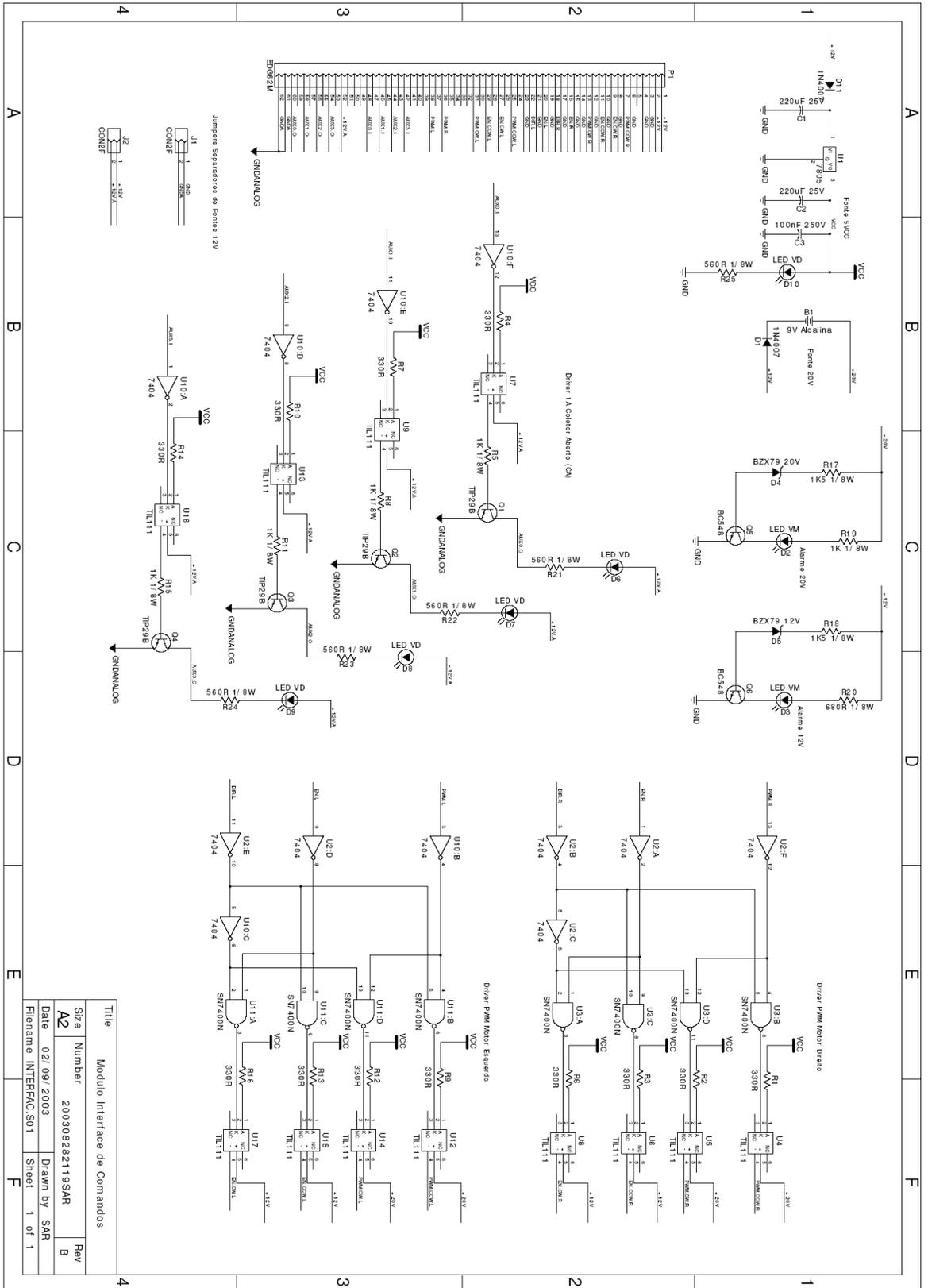


Figura 36. Esquemático PCI Interface do Sonic.

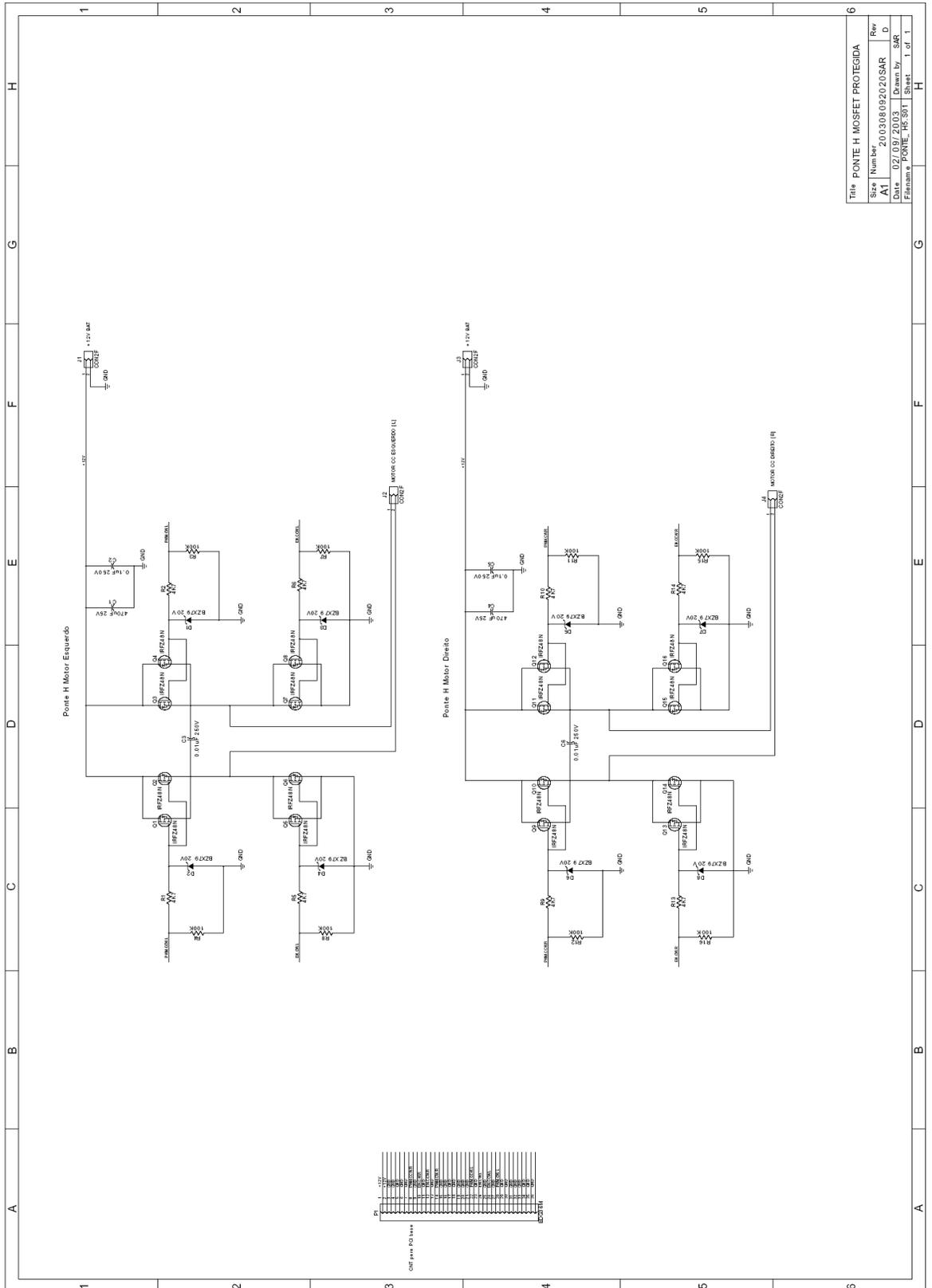
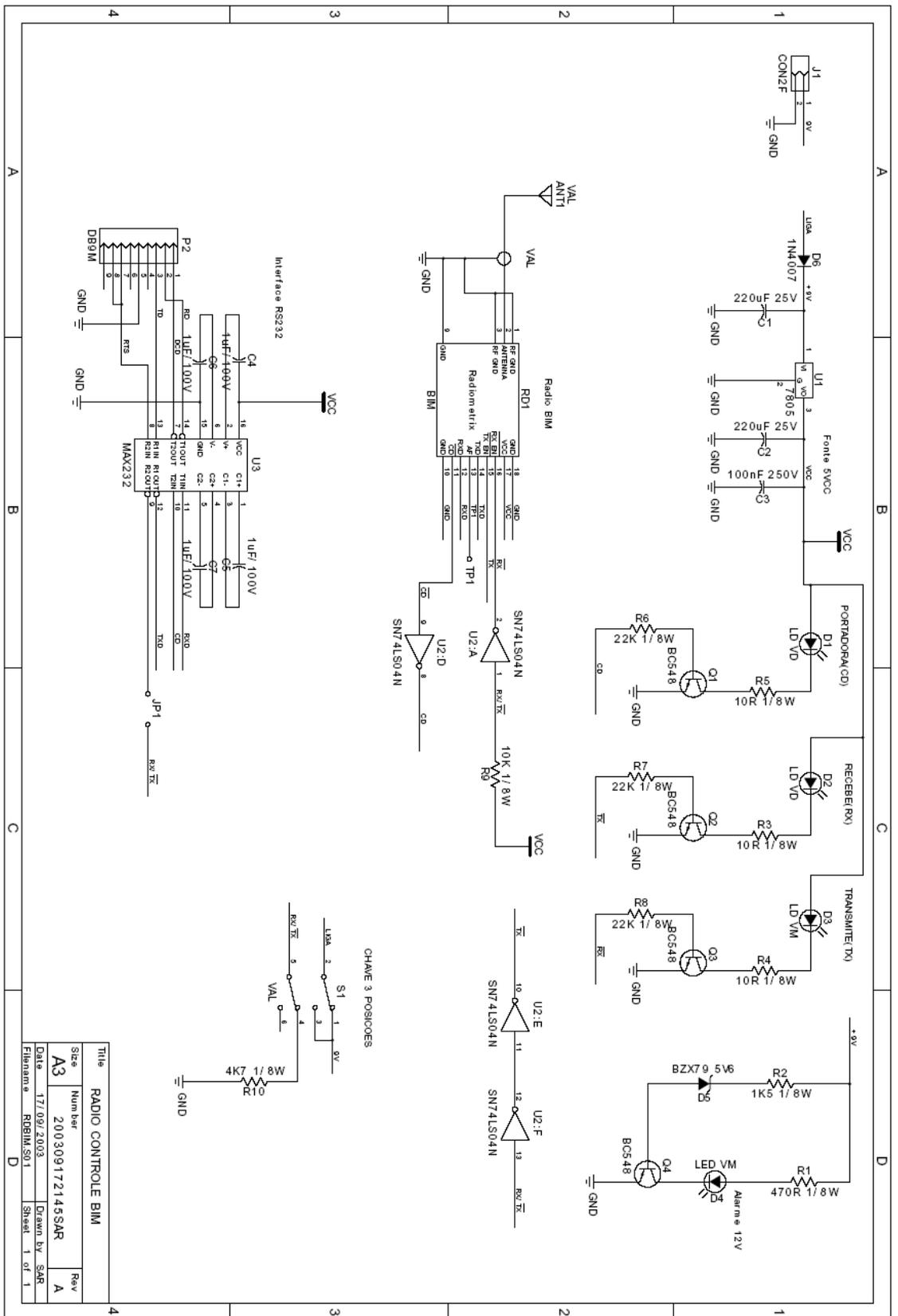


Figura 37. Esquemático PCI Ponte H.



Title	RADIO CONTROLE BIM	
Size	Number	Rev
A3	200309172145SAR	A
Date	17/09/2003	Drawn by
Filename	ROBIM.S01	Sheet
		1 of 1

Figura 38. Esquemático PCI Rádio Controle para a HP48.