

Desenvolvimento mecânico e de Controle PWM para sistema robótico

Ivo Takao Futida e Roseli Aparecida Francelin Romero

1. Introdução

Este relatório descreve as atividades desenvolvidas para a construção de um robô móvel e de um controlador em PIC, utilizando PWM (Pulse Width Modulation). O robô considerado aqui pertence a categoria “very small size” utilizado em times de futebol de robôs.). O controlador desenvolvido visa o controle dos motores, isto é, controla o valor da velocidade dos motores e assim possibilita que o robô faça curvas e mude de direção e aceleração. Este trabalho foi realizado junto à equipe de futebol de robôs do LAR – Laboratório de Aprendizado de Robôs, do ICMC-USP, Campus de São Carlos.

Serão descritos itens referentes ‘a construção do robô, desde a parte mecânica do robô, o desenho do projeto mecânico até a sua montagem (motores, chassi, carenagem, rodas, entre outros) bem como, as atividades referentes ao controle relativo à velocidade em cada um dos motores, promovendo o movimento do robô, uma vez que este apresenta apenas um eixo também são descritos.

Um estudo sobre a dinâmica do carrinho e os modelos de engenharia a serem aplicados na prática (que são diferentes dos ideais, devido às diversas condições do ambiente), também é apresentado.

O desenvolvimento do sistema de controle é de mais baixo nível, relacionado ao movimento do robô (velocidade). Para a realização desta etapa, foi aplicado o conhecimento de módulos do PIC e utilizando o recurso de PWM, realizou-se o controle de velocidade dos robôs do time de futebol de robôs USP Droids.

Esses robôs não possuem sensoramento interno, logo a aquisição da orientação vem diretamente do sistema de visão, através de diretivas calculadas pelo sistema de estratégia. Testes estão constantemente sendo realizados visando o melhor ajuste de velocidade gerado para os motores.

O presente relatório está organizado como segue. Na seção 1, são apresentados alguns conceitos introdutórios sobre futebol de robôs, sobre a evolução dos circuitos integrados, robôs móveis não-holonômicos e sobre o microcontrolador PIC. Na seção 2, são apresentados os vários projetos desenvolvidos para o robô protótipo até se chegar no projeto mecânico adotado. Nesta seção são apresentados também um estudo do microcontrolador PIC e detalhes da sua programação para a construção do sistema de controle e também da eletrônica. Finalmente, na seção 4, são apresentadas as conclusões sobre o trabalho realizado.

1.1 Futebol de robôs

Uma partida de futebol entre robôs autônomos, este foi o desafio lançado em 1996, por um grupo internacional de pesquisadores em Inteligência Artificial e Robótica Inteligente. O futebol de robôs reúne grande parte dos desafios presentes em problemas eminentemente distribuídos do mundo real, tais como, veículos autônomos, busca de informação em bases de dados distribuídos, planejamento da geração de energia elétrica, recomposição de linhas de transmissão, controle de tráfego aéreo e urbano, etc. Sendo assim, o futebol de robôs apresenta-se como um laboratório para pesquisa e ensino em automação e informática industrial.

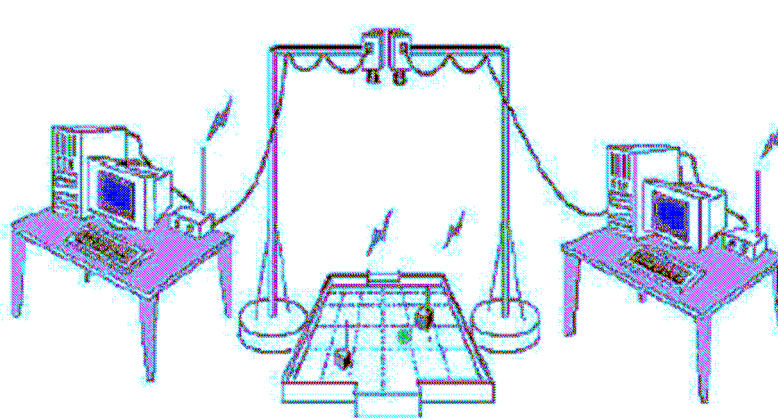


Fig. 1 Esquemático do jogo de futebol de robôs

Um time de futebol de robôs consiste em uma coleção de veículos autônomos capazes de reconhecer o ambiente onde estão inseridos, no caso o campo de futebol e seus pontos de referência, e os objetos pertencentes a este ambiente, a bola, os outros veículos que compõem o time e os adversários. Estes dispositivos devem ser capazes de representar o ambiente, estabelecer metas, planejar e executar ações para atingir tais metas. Em se tratando de um jogo de equipe, além do caráter autônomo, os jogadores de um time de futebol de robôs devem ser capazes de interagir com os outros jogadores do time para estabelecer objetivos coletivos, metas globais, planejar, alocar tarefas aos demais integrantes do time, sincronizar as ações de forma a imprimir ao time um perfil cooperativo. A construção de um time de futebol de robôs envolve a integração de diversas tecnologias, como projeto de agentes autônomos, cooperação em sistemas multiagentes, estratégias de aquisição de conhecimento, sistemas de tempo real, sistemas distribuídos, reconhecimento de padrões, integração de sensores.

Como podemos ver, o futebol de robôs é uma plataforma muito rica em termos de pesquisa e aprendizado. Esta é a visão que desejamos difundir nos principais centros de informação. As técnicas aplicadas neste ambiente podem ser utilizadas em outros ambientes com as devidas alterações, por exemplo o sistema de reconhecimento de objetos (visão), pode ser utilizado na indústria, por manipuladores que selecionam produtos. O sistema de estratégia, pode otimizar o caminho que um produto deve seguir numa linha de produção. Enfim, se conseguirmos promover o futebol de robôs, certamente o trabalho será de grande contribuição não somente para a Robótica em si, mas para as indústrias e outras plataformas de pesquisa.

1.2 Histórico

Em 1940, a computação era realizada por meio de válvulas (1ª geração). A segunda geração de computadores veio com os transistores (1958) e a 3ª geração data de 1964 com os Circuitos Integrados (SSI).

A evolução da computação nos últimos 50 anos se deu de forma impressionante. O advento dos transistores em 1958, que revolucionou a eletrônica, contribuiu de forma significativa para o aperfeiçoamento do processamento nos computadores e equipamentos eletrônicos.

Uma das ramificações deste aperfeiçoamento na eletrônica deu origem aos multiprocessadores RISC (“Reduced Instruction Set Computer”), dos quais, o

microcontrolador PIC é um exemplo de aplicação (com tecnologia CMOS), seu conjunto de instruções RISC se compõe de 35 instruções.

Até 1965, somente eram disponíveis semicondutores encapsulados individualmente. Os engenheiros montavam portas e sistemas digitais a partir destes componentes individualmente encapsulados e de resistores. Dispositivos semicondutores são fabricados através da aplicação repetida de certos processos físicos e químicos à superfície de uma pastilha de silício extremamente puro.

Os dois detalhes da fabricação de semicondutores não nos interessam exceto no que tange às dimensões do dispositivo na superfície do silício, que são da ordem de microns. A maior parte da dimensão de um dispositivo encapsulado individualmente envolve não o dispositivo em si, mas o próprio encapsulamento e o suporte mecânico necessário para as conexões elétricas.

Assim iniciou-se uma série de progressos tecnológicos (que prosseguem sem que se possa vislumbrar seu final) que levaram a fabricação de um dispositivo semicondutor chamado *circuito integrado* (CI). Em um circuito integrado, muitos transistores e diodos são fabricados, isto é, integrados, sobre a mesma pastilha de silício; na mesma estrutura são também integrados os resistores e até mesmo as interligações necessárias para fabricar uma porta completa, muitas portas e, até mesmo, um sistema digital completo.

No caso do projeto de futebol de robôs (categoria mirosot), parte do processamento é realizada no PIC, que é a modulação da largura de pulso (PWM). Embora sejam independentes uns dos outros, todo o processamento restante é efetuado num computador central, baseado em informação (posição dos jogadores, da bola, etc.) vinda de uma câmara de vídeo colocada por cima do campo de jogo, os comandos são enviados para os robôs via *wireless* (rádio). A intervenção humana resume-se à introdução e remoção dos robôs no campo.

1.3 Robôs Móveis Não-Holonômicos

Os robôs móveis e, em particular, os robôs terrestres têm sido muito estudados nos últimos anos. Para se locomoverem, estes robôs podem se servir de rodas, esteiras, “patas” ou outros métodos de locomoção. Neste trabalho nos concentraremos nos robôs móveis dotados de rodas. Uma característica importante dos robôs móveis dotados de rodas é a presença de restrições não-holonômicas. Uma restrição não-holonômica impede que o robô execute movimentos normais à superfície do corpo de suas rodas. A Figura 1.2 apresenta com clareza como as restrições não-holonômicas se apresentam em uma aparentemente simples tarefa de estacionamento.

Sem levar em consideração o sistema de locomoção, seria uma linha reta deslocando o robô lateralmente para a posição final desejada (a). No entanto, esta trajetória não é realizável por este sistema de locomoção. Desta forma, é necessária uma trajetória levando em consideração as restrições não-holonômicas impostas pelo sistema de locomoção (b). [13]

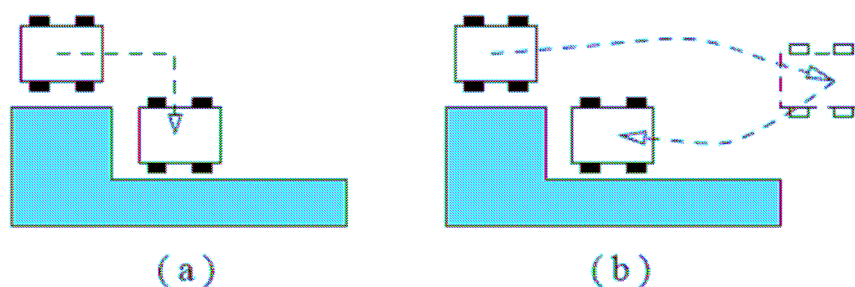


Figura 2 - Estacionamento de um robô móvel com restrições não-holonômicas. Em (a) o movimento não executável devido às restrições e em (b) o movimento executável

1.4 Microcontroladores

Os microcontroladores são componentes utilizados em praticamente qualquer equipamento eletrônico digital. Cerca de 50% dos microcontroladores vendidos são controladores "simples", outros 20% são processadores de sinais digitais (DSPs) mais especializados.

Os microcontroladores podem ser encontrados em praticamente todos os dispositivos eletrônicos digitais que nos cercam: teclado do computador, dentro do monitor, disco rígido, relógio de pulso, rádio relógio, máquinas de lavar, forno de microondas, telefones, etc. Certamente eles foram tão ou mais importantes para a revolução dos produtos eletrônicos que os computadores. Eles permitiram a evolução de equipamentos que a anos não evoluíam, como os motores a combustão, que agora com o novo controle eletrônico podem funcionar com sistema bicomcombustível e poluindo menos.

Para a execução do projeto foi utilizada a linguagem Assembly, que apesar de ser de baixo nível, exige do programador maior proximidade com os recursos disponíveis no microcontrolador: conhecimento e otimização dos espaços para memória, configuração de registradores, aplicações com contadores, interrupção.

1.5 PIC 16F877A

O microcontrolador PIC apresenta estrutura de máquina interna do tipo Harvard, enquanto a maioria dos microcontroladores apresenta arquitetura Von-Neumann. A diferença está na forma como os dados e o programa são processados pelo microcontrolador. Na arquitetura tipo Harvard, existem dois barramentos internos, sendo um de dados e o outro de instruções. O barramento de dados é de 8 bits e o OP-CODE é de 14 bits.

O PIC 16F877A possui 35 instruções (número razoavelmente reduzido) e por este motivo é classificado como tecnologia RISC- Reduced Instruction Set Computer. Isso torna o aprendizado mais fácil e dinâmico, mas por outro lado, implica no fato de

que muitas funções devem ser “construídas”, pois não existe uma instrução direta, exigindo maior habilidade do programador. [12]

O microcontrolador é programável porque toda lógica de operação é estruturada na forma de programa e gravada dentro do componente. Assim, toda vez que o microcontrolador for alimentado, o programa interno será executado. Quanto à inteligência do componente, podemos associar à Unidade Lógico Aritmética (ULA), pois é nesta unidade em que todas as operações matemáticas e lógicas são executadas. Quanto mais poderosa a ULA do componente, maior sua capacidade de processar informações.

2. Projeto do robô protótipo

A Figura 3 mostra o primeiro protótipo desenvolvido para o robô do time de futebol de robôs ‘USPDroids’, que era basicamente um ‘carro’ experimental, pouco eficiente e simples, consistindo de uma carcaça de nylon preto, que é um material de fácil manuseio. A transmissão do eixo do motor se dá através de uma coroa, tendo uma redução próxima de 2:1, baixa para a aplicação em questão. A prática mostrou que era necessário algo mais robusto e resistente.

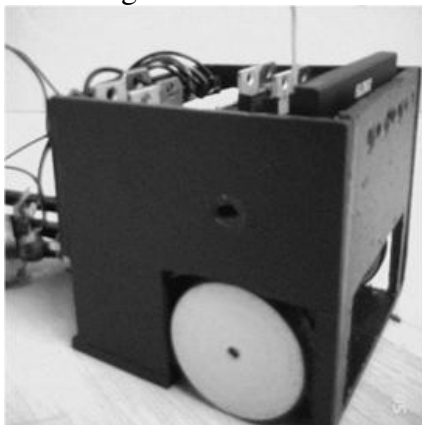


Fig. 3 Protótipo experimental

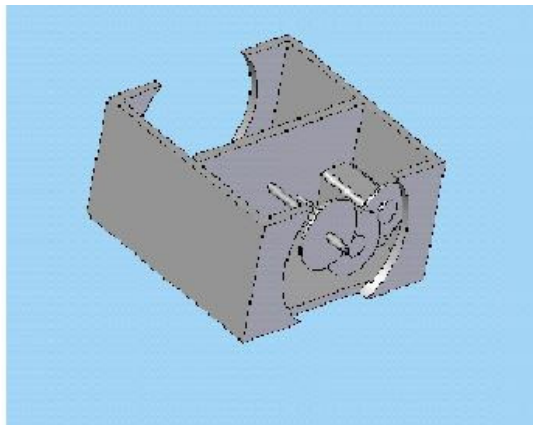


Fig. 4 Esboço da primeira caixa de redução

O robô necessita de um corpo para acoplar todos os seus elementos (motores, placas, baterias), que protege e realiza choques com oponentes/bola no jogo de futebol de robôs. Encontramos então, a primeira dificuldade: a de projetar uma caixa de redução ou um arranjo com motor com redução para a construção do robô protótipo.

O material escolhido foi o Alumínio (aeronáutico), pois este é um dos metais que possuem menor densidade específica (aproximadamente 1/3 da densidade dos aços), comporta-se bem como isolante elétrico e não gera problemas de contato com a placa/eletrônica e, sobretudo apresentam uma relação rigidez/resistência ótima. Este material pode ser trabalhado com usinagem, durante o processo de fabricação da peça, ao contrário do nylon preto, por exemplo, que em situação semelhante não suportaria determinadas tensões de cisalhamento.

Nos ensaios, verificou-se que esse material tornou a caixa relativamente pesada e que apesar disso, não foi um empecilho sério para o projeto, uma vez que é interessante que os robôs tenham uma massa significativa para não ter problemas de impacto entre robôs.

1ª solução

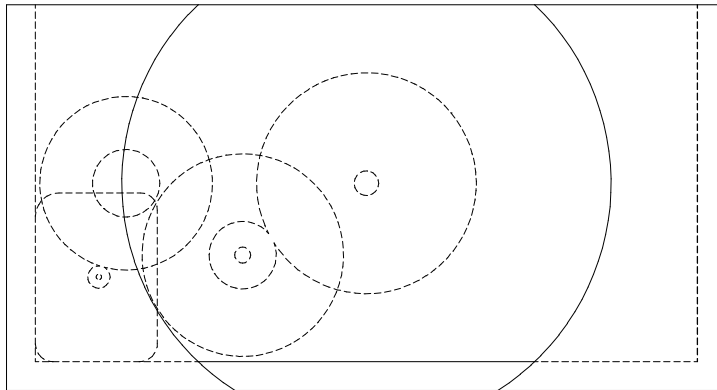


Fig. 5 Vista lateral: engrenagens, motor e eixo da roda

A primeira solução implementada é mostrada nas Figuras 3 e 4. Projetamos uma caixa com 3 engrenagens, de forma que a engrenagem final está fixa no próprio eixo da roda.

Surgiram muitos erros de medida, de paralelismo e perpendicularidade, de fixação dos eixos nas paredes da caixa, de folga entre as engrenagens. A caixa é “montada” a partir de uma chapa de Alumínio de 1.5mm de espessura, que têm suas medidas desenhadas na chapa e então é cortada com serra; posteriormente as paredes são dobradas. Portanto as próprias paredes da caixa apontavam a falta de paralelismo entre as paredes e falta de rigidez.

A própria montagem dos elementos da caixa torna essa solução difícil e de manutenção contínua. Os resultados foram importantes para verificar a necessidade de uma estrutura mais firme e que possibilite o encaixe de eixos e possivelmente, rolamentos.

A forma de alimentação do sistema consistia em 8 pilhas AA 1.2V 2300mAh, dos quais 4 alimentam os motores e 4 o sistema eletrônico.

2ª solução

Então, a segunda conjectura para o protótipo foi desenvolvida pelo bolsista. Simulando com o software de desenho, esse projeto trata com maior precisão as medidas, uma vez que a ‘caixa’ do robô passa por processo de produção distinto. Com o recurso de usinagem, a peça pode ser manipulada a partir de um bloco do material selecionado (alumínio) e então, através do ferramental adequado (que é o limitante nesse projeto), dar a forma desejada.

A grande vantagem nesse caso é a homogeneidade e rigidez estrutural da caixa. Projetando conforme a necessidade, os motores são facilmente acoplados, os furos para os eixos são mais precisos e há mais espaço para um projeto elaborado com furos roscados, rolamentos para reduzir o atrito no eixo.

O desenho técnico mecânico feito no software Solid Edge da caixa se encontra abaixo:

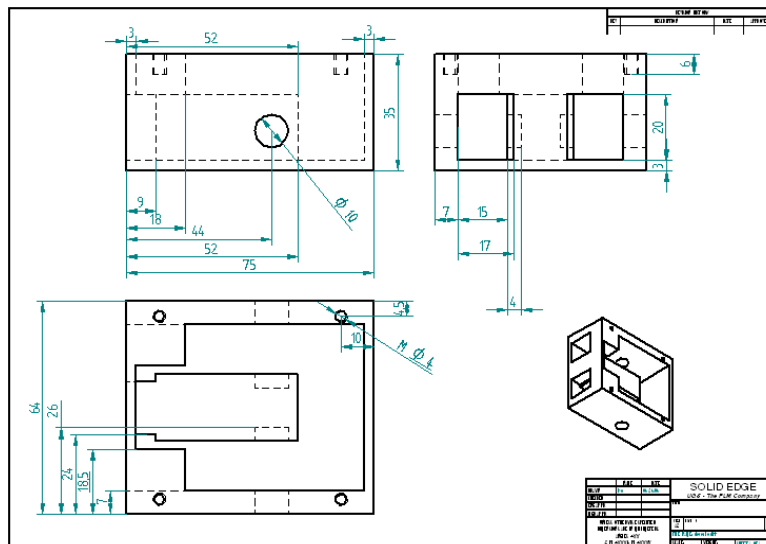


Fig. 6 Desenho Técnico Mecânico da caixa de redução feito no Solid Edge Draft

O sistema completo pode ser visto na Figura 7, em que o bloco superior corresponde a parte eletrônica.

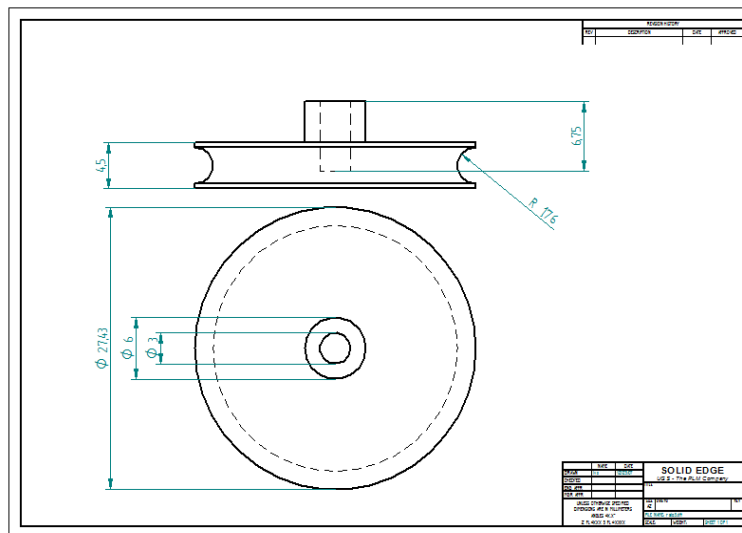


Fig. 7 Desenho Técnico Mecânico da roda do robô

A roda foi projetada para suportar externamente o anel O-ring de diâmetro interno 26 mm e externo de 28 mm. A escolha desse anel é justificada para maior aderência ao piso do campo de futebol de robôs, pois quanto menor o escorregamento das rodas, menor será o erro propagado sobre o sistema de controle. Basicamente esta se prende ao eixo através de araudite.

O rolamento utilizado tem diâmetro externo de 10 mm e diâmetro interno de 3 mm, no total são acoplados quatro rolamentos, dois para cada eixo. O eixo é um pino de aço com 2,8mm de diâmetro. A transmissão do movimento do eixo do motor é realizada pela mesma coroa da solução 1. A Figura 5 mostra os detalhes do projeto, o encaixe dos motores se dá perfeitamente, para evitar que este movimento durante a vibração do sistema.

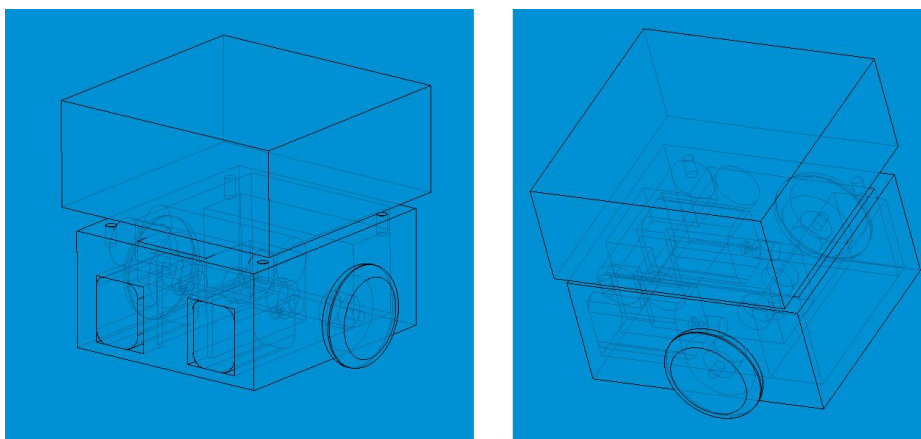


Fig. 8 Vistas isométricas do robô virtual

Este projeto se baseou nas medidas do motor alternativo que encontramos, (não havia dados disponíveis sobre o motor), o diâmetro do eixo/rolamento e nas dimensões da bateria.

A usinagem foi então realizada por uma empresa especializada (terceirizada), que através do maquinário moderno: torno, fresa (máquinas CNC) principalmente, trabalhou a peça com perfeição.

A parte de redução de atrito/ruído deste projeto melhorou significativamente. Observe que projetamos a caixa para acoplarmos quatro rolamentos, o que evita que o eixo perca torque através do atrito, como ocorria no projeto anterior.

A alimentação do sistema mudou devido ao volume que as pilhas tomavam no projeto anterior. Optamos por baterias de lítio, que possuem uma voltagem maior, podem ser recarregadas e uma maior capacidade de energia armazenada por volume em comparação com as demais disponíveis no mercado. Por seu ótimo desempenho, esse tipo de bateria é também bastante utilizado em aeromodelos, onde a busca por redução de peso e volume também é importante para o desempenho do mesmo.

Tabela 1 – Dados do sistema de alimentação

Bateria	Tensão (V)	Amperagem (mAh)
Hi-Model Litihium Polimer rechargeable cells	7,4	1300
GP rechargeable NIMH	8,4	170

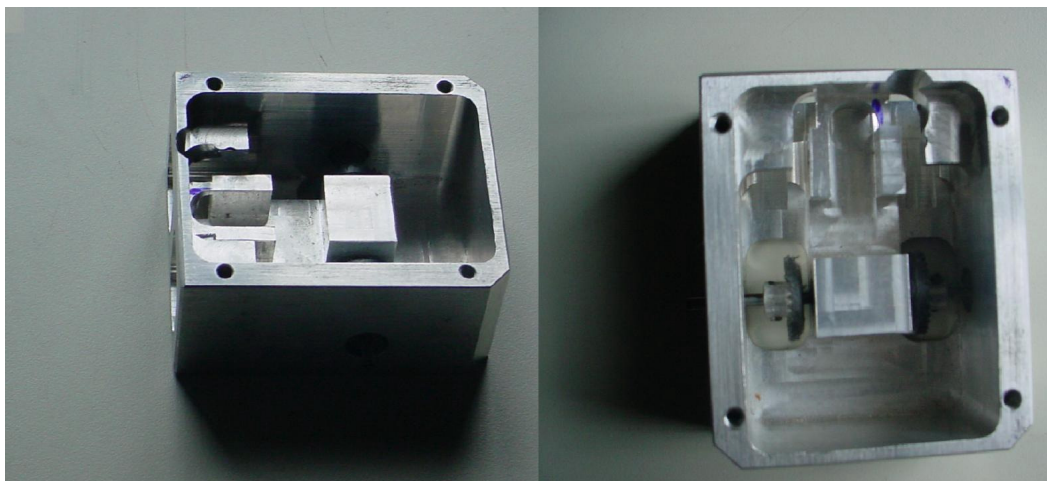


Fig.9 Fotos da caixa projetada usinada; com eixo, pinhão e rolamentos instalados

Apesar de todos os detalhes do projeto, ao realizarmos os experimentos, verificamos que o motor DC e a redução realizada pelo pinhão escolhido não eram suficientemente potentes para fornecer o torque desejado. A velocidade nas rodas era alta, porém não possuía torque suficiente para mover o protótipo quando estava sob força do próprio peso. A caixa ficou mais pesada do que o esperado, mesmo retirando parte considerável do peso do robô, o projeto foi inviável.

Considerando que a relação entre potência e velocidade angular de um motor de corrente contínua é praticamente uma função linear que passa pela origem do plano cartesiano; a baixas rotações, a potência entregue ao sistema é mínima, necessitando de mais redução mecânica para que o movimento circular (nos eixos) seja convertido para movimento linear (movimento do robô em relação ao plano).

A dificuldade na parte de montagem da caixa foi outro fator agravante do projeto. A junção entre a coroa e o pinhão do eixo do motor apresentava erros de geometria, de forma que em alguns instantes o eixo travava.

Tendo em vista que os erros de projeto estavam agravados pela escolha do motor, a primeira decisão a ser feita foi mudar o motor e a adaptação da carcaça do robô para um projeto mais simples e de fácil montagem, tal que pudéssemos realizar no próprio laboratório de metrologia Lamafe – USP.

3ª solução

Nesse processo, escolhemos os Motores DC que possuem controle preciso de velocidade e são mais disponíveis e baratos no mercado.

Das especificações do motor, fornecida pela Action Motors, temos:

Tabela 2 - Dados do motor PM020-S

MODELO	TENSÃO (V,dc)		EM VAZIO		MÁXIMO RENDIMENTO					BLOQUEIO	
			Rotação	Corrente	Rotação	Corrente	Torque	Potência	Rendimento	Torque	Corrente
	Operação	Nominal	RPM	A	RPM	A	g.cm	W	%	g.c.m	A
PM020-S	1,4-3	3	8500	0,08	5800	0,17	2,7	0,16	32,4	8,5	0,35

Para este último projeto usamos uma engrenagem com diâmetro quatro vezes maior que o diâmetro do pinhão do eixo do motor (redução 1:4). A roda usinada em nylon preto, com dois anéis O-ring (para aumentar a aderência com o chão) possui 50 mm.

A caixa é composta por paredes que são ‘montáveis’ por meio de parafusos M3.

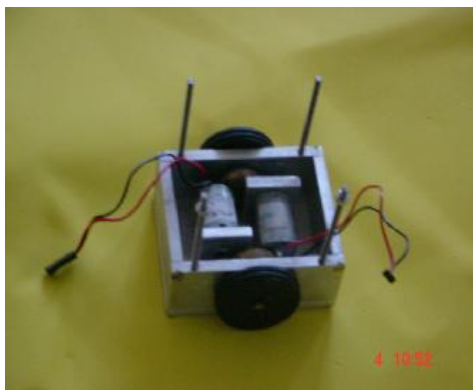


Fig. 10 Protótipo montado



Fig. 11 Vista superior

O sistema de alimentação permaneceu o mesmo da solução 2.

O eixo prende a roda através de duas presilhas de aço.

Os pilares M3 servem para suportar a parte eletrônica e são rosqueados à carenagem.

Posteriormente foi feita uma carenagem para o robô, de polycarbonato, com uma chapa de 2,5 mm de espessura, para que esta englobasse o robô inteiro e assim pudesse finalizar o corpo do protótipo, dando maior proteção à parte eletrônica. A montagem da carenagem foi realizada basicamente por colagem com araudite.

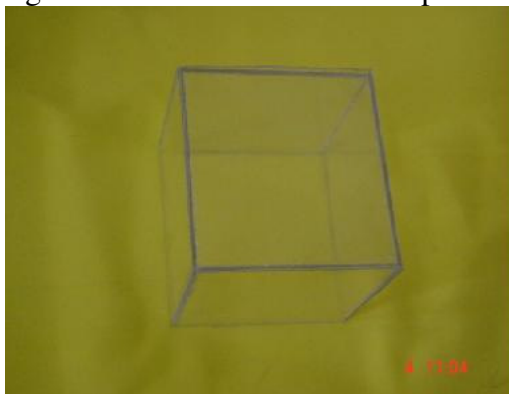


Fig. 12 Carcaça de Polycarbonato



Fig. 13 Protótipo completo

Nos testes, esta carenagem se mostrou muito frágil; devido aos choques entre os robôs e as paredes do campo de futebol. A baixa resistência oferecida pelo processo de montagem provocou a quebra da carenagem durante alguns testes. Para a competição de outubro, durante a Competição de Robótica Brasileira CBR2007 no VIII SBAI - Simpósio Brasileiro de Automação Inteligente, a carenagem foi substituída por espuma.

Alterar o formato da carcaça do robô, diminuir as dimensões da roda, são algumas das observações que poderiam estar incluídas num novo projeto.

Mas para efeito de testes e calibração, essa nova caixa mostrou-se suficiente e bastante robusta com relação a todas as outras. O protótipo foi construído com sucesso e a parte de controle pode ser desenvolvida a partir do mesmo.

3.1 Projeto Mecânico

Na parte mecânica referente ao primeiro relatório, o projeto mecânico final não foi ilustrado. Aqui se encontram as especificações detalhadas da montagem do protótipo: eixo, roda, engrenagens, paredes da caixa de redução e um desenho com a montagem completa do protótipo, através do software Solid Edge versão acadêmica.

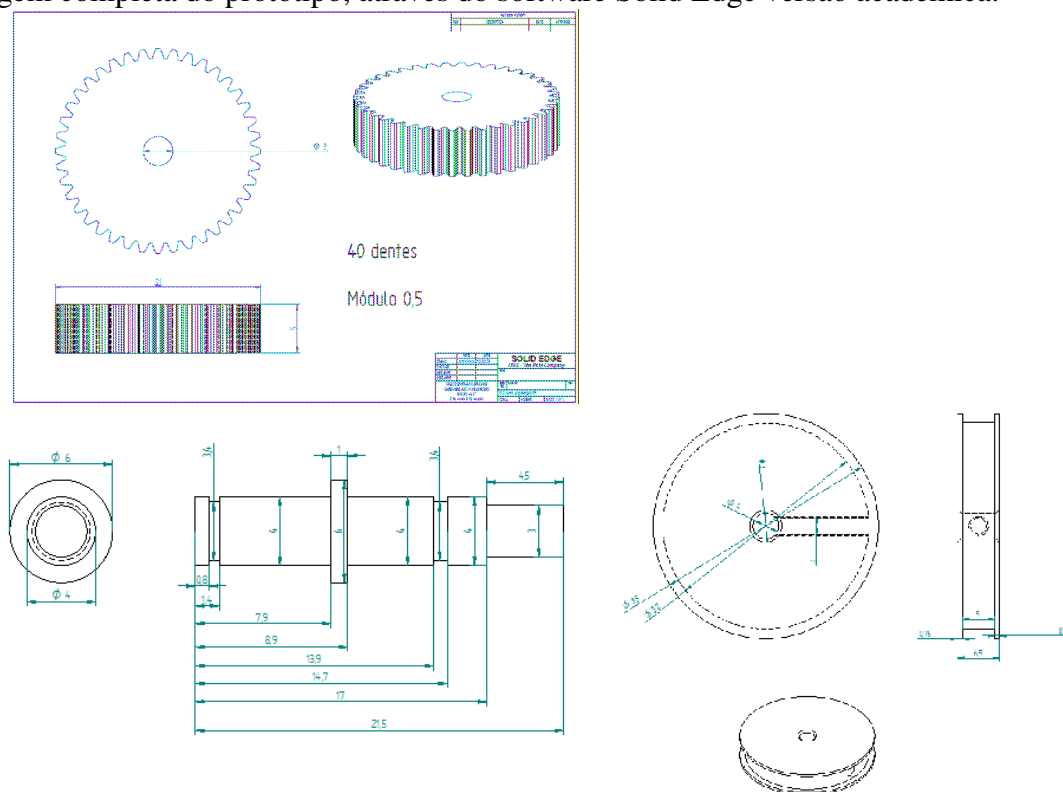


Figura 14 – Desenho técnico mecânico do eixo principal e da roda de nylon

Tabela 3 – Relação de acoplamento de peças ao eixo

Elemento	Forma de Acoplamento ao eixo
Rolamentos	Ajuste interferente (eixo-base)
Engrenagens	Chaveta externa
Rodas	Parafuso M3

Para garantir a fixação da roda e rolamentos, presilhas são colocadas nos orifícios do eixo. A roda depois de parafusada é ‘preenchida’ com dois anéis O’ring, que aumentam a adesão com o chão.

As paredes (de Alumínio aeronáutico) são presas umas as outras por meio de parafusos M3 com cabeça escareada.

Através do software Solid Edge Assembly, o protótipo foi montado e uma simulação do motor foi feita, observando possíveis erros de interferência, que foram corrigidos, vide Figura 16.

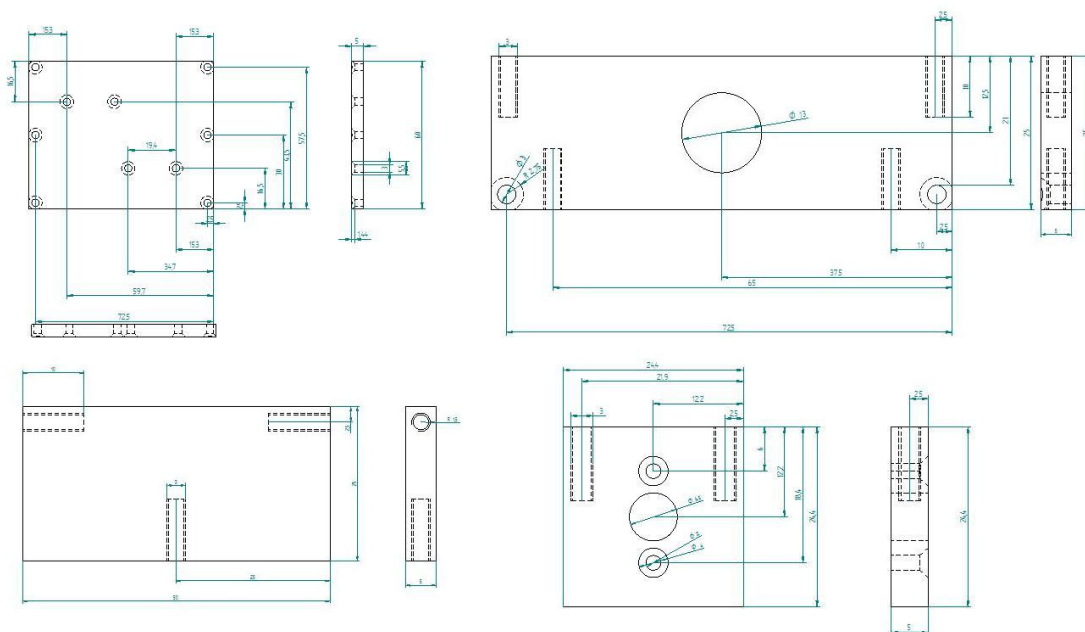


Figura 15– Desenho técnico da caixa de redução a ser montada

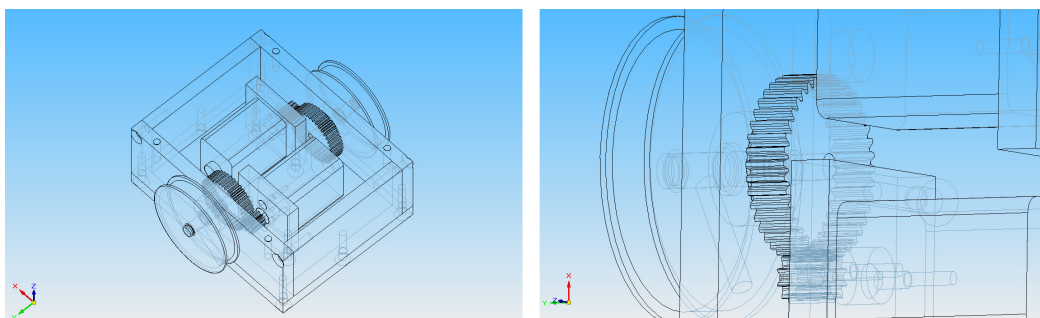


Figura 16-a) Montagem virtual do carro b) Conexão de engrenagens com o eixo

A limitação de espaço na categoria Very Small Size de futebol de robôs [11], é bastante evidente, o robô que deve ter um tamanho máximo de um cubo com 75mm de aresta. Por esse motivo, não há espaço para um motor mais robusto, que tenha acoplado um encoder e/ou redução. A engrenagem, mostrada nas Figuras 3 e 4, foi projetada e usinada. Esta peça é a que demanda mais tempo para usinagem e conseqüentemente mais custo.

3.2 Desenvolvimento do Sistema de Controle / Estudo da programação em PIC

O PIC16F877A possui 40 pinos e uma gama de recursos disponíveis:

- Microcontrolador de 40 pinos, o que possibilita a montagem de um hardware complexo e capaz de interagir com vários recursos e funções ao mesmo tempo;
- Via de programação com 14 bits e 35 instruções;
- 33 portas configuráveis como I/O;
- 15 interrupções disponíveis;
- Memória de programação E²PROM FLASH;

- Memória E²PROM (não-volátil) interna com 256 bytes
- Memória RAM com 368 bytes
- 3 Timers
- Comunicações Seriais
- **2 Módulos CCP: Capture, Compare, PWM**

O modo PWM é um dos recursos mais poderosos dos módulos CCPs, pois com ele podemos obter uma tensão analógica a partir de um sinal digital (que assume apenas os estados 0 ou 1).

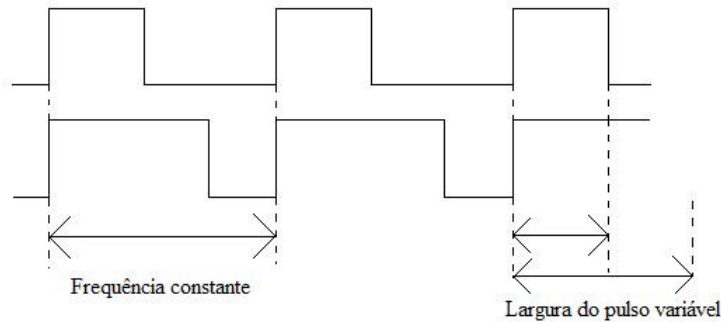


Fig. 17 Esquema da modulação da largura de pulso

Da teoria, sabemos que, a tensão média de uma forma de onda é dada por:

$$V_{dc} = \frac{1}{T} \int_0^T V(t) dt$$

Onde T é o período da forma de onda e V(t) é a função da tensão no tempo.

$$V(t) = \begin{cases} V_{pulso} \rightarrow 0 \leq t \leq t_p \\ 0 \rightarrow t_p < t \leq T \end{cases}$$

t_p : duração do pulso em nível lógico 1

V_{pulso} : tensão de pulso do sinal PWM

Então,

$$V_{dc} = \frac{1}{T} \left(\int_0^{t_p} V_{pulso} dt + \int_{t_p}^T 0 dt \right)$$

$$V_{dc} = \frac{t_p}{T} V_{pulso}$$

A razão entre a largura de pulso e o período da forma de onda recebe o nome de *duty cycle*. O pulso da onda PWM apresenta tensão fixa, porém o valor médio da tensão desta forma de onda varia em função do *duty cycle*. A tensão média (V_{dc}) é diretamente proporcional ao *duty cycle* e como esse varia entre 0 (quando $t_p=0$) e 1 (quando $t_p=T$)

temos que a tensão média de onda pode variar entre 0 e V_{pulso} . Neste caso (PIC16F877A), a variação será de V_{ss} a V_{DD} , ou seja, de 0 a 5V. [1]

Vejamos a nomenclatura padrão utilizada para descrição das características comuns dos módulos CCP do PIC: [7]

Tabela 4 - Características comuns dos módulos CCP do PIC

Padrão	CCP1	CCP2	Descrição
CCPxCON	CCP1CON	CCP2CON	Registrador de configuração
CCPRxH	CCPR1H	CCPR2H	Parte alta do valor de controle
CCPRxL	CCPR1L	CCPR2L	Parte baixa do valor de controle
CCPx	CCP1(RC2)	CCP2(RC1)	Pino relacionado

Esses dois módulos utilizam recursos compartilhados para suas bases de tempo (Timer 1 e Timer 2).

Tabela 5 - Combinação de recursos dos módulos CCP

Modo	Base de tempo
Capture	Timer 1
Compare	Timer 1
PWM	Timer 2

Como utilizaremos as 2 portas CCP com o recurso PWM, ambos os PWM terão a mesma frequência e serão sincronizados, devido ao uso da mesma base de tempo. Os ‘Duty Cycles’ possuem controles independentes.

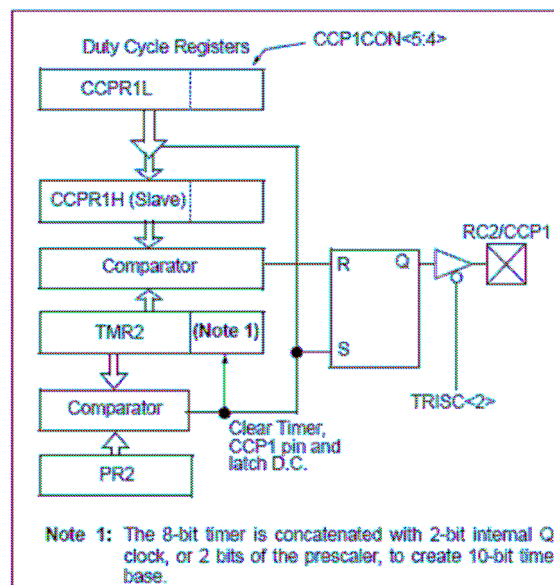


Figura 17 – Diagrama de blocos do módulo CCP no modo PWM [12]

Configuração para a operação do PWM:

1. ConFigurar o período do PWM escrevendo-o no registrador PR2
O período T do PWM é controlado diretamente pelo Timer 2, através do registrador PR2.
O período do PWM pode ser calculado usando a fórmula:

$$\text{PWM Período} = [(PR2) + 1] \cdot 4 \cdot T_{osc} \cdot (\text{Valor TMR2 Prescale})$$

Onde a frequência do PWM é definida como $1/[\text{PWM período}]$.

2. Configurar o PWM *duty cycle*

No PIC, não definimos o valor do *duty cycle*, mas sim, do tempo do pulso em nível alto, que pode ser calculado da seguinte maneira:

$$t_p = CCPRxL:CCPxCON<CCPxX:CCPxY> \cdot T_{osc} \cdot (\text{Prescale do TMR2})$$

3. Fazer do pino uma saída zerando o bit TRISC<2>

Na modulação PWM, o pino CCPx produz uma saída PWM com 10 bits de resolução. Como o pino CCP1 é multiplexado com o PORTC data latch, o bit TRISC<2> deve ser zerado de forma que faça do CCP1 um pino de saída.

Quando TRM2 = PR2, ocorrem os seguintes eventos:

- TMR2 é zerado, iniciando novo período;
- O pino CCP1 é ‘setado’ (exceção: se o PWM *duty cycle* = 0%)
- O PWM ‘*duty cycle*’ é copiado do CCPR1L para o CCPR1H

4. Configurar o valor TMR2*prescale*, habilitar o Timer2 via T2CON

A resolução para o ajuste de período depende do *prescale* do Timer 2:

Tabela 6 - Influência do Prescale no tempo de pulso em nível alto

Prescale	Tempo do menor passo
1	T_{osc}
4	$4T_{osc}$ OU T_{cy}
16	$16T_{osc}$ OU $4T_{cy}$

5. Configurar o CCP1 para o modo de operação PWM.

Para ativar a saída do módulo de PWM setamos os bits CCPxCON<CCPxM3:CCPxM0>

3.2.1 Módulo Eletrônica

A comunicação é a parte responsável pela interligação dos robôs ao módulo de estratégia via rádio frequência, é unidirecional por possuir somente um sentido (estratégia para robôs), e é constituída de duas componentes:

- recepção de informações: componente conectada ao robô para recepção das informações. Esse módulo recebe o sinal serial pela antena e envia essas informações para o PIC, de forma serial.
- envio de informações: componente conectada a um computador executando o software de estratégia.

A recepção é feita através do circuito RXM-900-HP3-PPO. A transmissão é feita através de um módulo de transmissão ligado a saída serial do computador que utiliza o circuito transmissor de rádio frequência TXM-900-HP3-PPO, um transmissor que opera na faixa de 902 a 928 MHz.

O conjunto RXM-900-HP3-PPO - TXM-900-HP3-PPO são os componentes principais de todo o módulo de comunicação. Eles são circuitos de fácil utilização, boa relação custo/benefício e permitem uma taxa de 56 Kbps de transmissão.

Escolhemos a comunicação serial, pois apesar de mais lenta que a paralela, possui um menor grau de interferência.

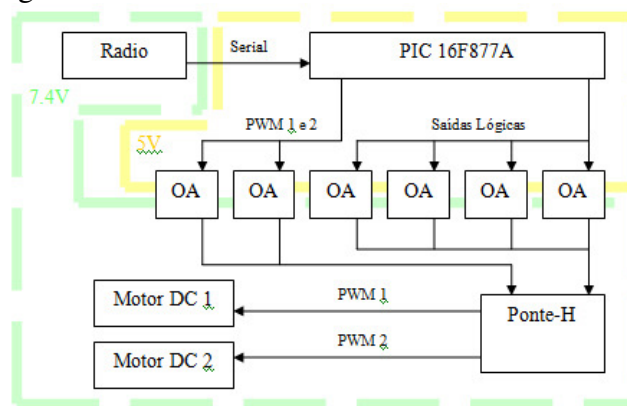


Fig. 18 Módulo eletrônico do robô com opto acopladores e duas baterias.

O time utiliza dois circuitos alimentados por fontes totalmente independentes que são interligados através de acopladores ópticos. Essa divisão em dois circuitos foi feita para separar o circuito de motores do circuito de controle devido ao alto consumo de energia de motores elétricos. Se fosse implementado somente um circuito, o alto consumo de energia pelos motores poderia fazer com que o nível de energia fornecido ao 16F877A ficasse abaixo do recomendando, causando um *reset* no 16F877A levando a perda de toda informação interna, um efeito indesejado que pode fazer com que o robô não obedeça corretamente as instruções enviadas pelo módulo de estratégia.[8]

Os parâmetros do sentido das rodas passam por uma lógica e se dividem em 4 portas do PIC, acionando os pinos de sentido da ponte-H. Já as velocidades das rodas acionam o mecanismo PWM do microprocessador e partem dois pulsos de portas diferentes, também em direção à ponte-H. São esses dois pulsos, associados aos quatro pinos lógicos de sentido que realizam a comunicação com o módulo da mecânica. A ponte H (módulo de potência) permite não só o comando instantâneo de dois motores por meio de sinais PWM, como também amplifica o sinal enviado pelo microcontrolador.

3.2.2 Programação do microcontrolador

O ambiente usado na programação foi o MPLAB, ferramenta da própria empresa MICROCHIP, na qual o programa é compilado em linguagem Assembly para um arquivo hexadecimal, o qual é gravado no PIC.

Basicamente, o controle de velocidade é localizado no próprio microcontrolador, que através do PIC, recebe o sinal via rádio da velocidade que deve ser aplicada as rodas. Essa velocidade é calculada pelo sistema de estratégia que é processada num computador. O PIC recebe essa informação e joga uma tensão modulada no pino de saída, de forma que essa tensão nos terminais do motor, produz a velocidade desejada.

Para realizar a aplicação de uma dada tensão através de um sinal, testes foram feitos na configuração do PIC. Conforme vimos no primeiro relatório, o módulo CCP será uma ferramenta de grande valor na modulação da largura de pulso.

PWM é utilizada para variar o valor da transferência de potência entregue a uma carga sem as perdas ocorridas normalmente devido à queda de tensão por recursos resistivos. Em um sistema PWM, a chave de estado sólido usada para controlar o fluxo de corrente: ora não conduzindo corrente, ora conduzindo, mas provocando uma queda de tensão muito baixa. Como a potência instantânea dissipada pela chave é o produto da corrente pela tensão elétrica a um dado instante, isso significa que nenhuma potência é dissipada se a chave fosse uma chave "ideal".

3.2.2.1 Etapa de configuração e cálculo

A título de exemplificação, tratamos uma modulação de largura de pulso através do PIC. Calculando o *duty cycle* do Pulso PWM em quatro situações, usando o Timer2, temos:

$$t_p = \text{CCPR2L:CCP2CON} < \text{CCP2X:CCP2Y} > * T_{\text{osc}} * (\text{Prescale do TMR2})$$

Utilizando um oscilador externo de 4MHz:

$$T_{\text{osc}} = \frac{1}{f_{\text{osc}}} = \frac{1}{4 \times 10^6} = 250\text{ns}$$

Definindo o prescaler do Timer2 em 16 (tempo do menor passo, com maior resolução), os bits do registrador CCP2CON, com os bits 5 e 4 (menos significativos) zerados e PR2 no valor máximo(255), temos que a frequência do PWM será de 244,14Hz ($T_{\text{PWM}} = 4,096\text{ms}$), através dos cálculos temos:

Tabela 7- Relação entre os valores adotados e porcentagem de *duty cycle*

CCPR2L(hexadecimal)	t_p (ms)	Duty Cycle (%)	Tensão (V)
0	0	0	0
0X80 (512)	2,048	25,0	1,45
0XC0 (768)	3,072	75,0	3,97
0XFF (1023)	4,072	99,9	4,81

A tabela 7 também apresenta a tensão obtida, pela leitura do voltímetro, da voltagem do pino do PIC com relação ao nível lógico baixo.

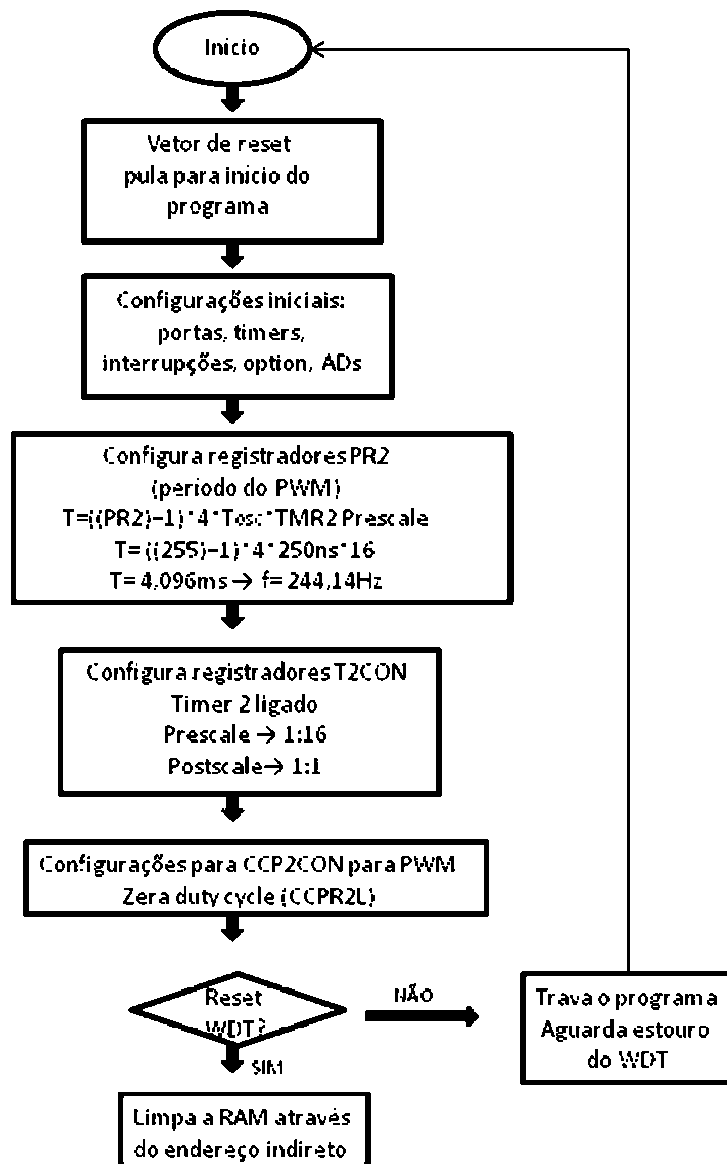


Figura 19- Fluxograma do programa para variação da tensão via PWM

A Figura 19 apresenta através de um fluxograma, as etapas de configuração do microcontrolador para a utilização do mesmo na etapa de testes:

Com o registrador PR2 com o valor de 255, o tempo máximo de pulso alto é no máximo 1023 enquanto a resolução do *duty cycle* é de 10 bits ($1024=4 \times (255+1)$), ou seja, nunca será atingido um *duty cycle* de 100%.

Para calcular os valores de CCPR2 é necessário aplicar os valores em:

$$CCPR2 = [(PR2)+1] \times 4 \times \text{Porcentagem de duty cycle}$$

O *Watchdog Timer* é uma espécie de temporizador que reseta o microcontrolador com base no tempo. Esse contador (WDT) é independente do ciclo de máquina do PIC.

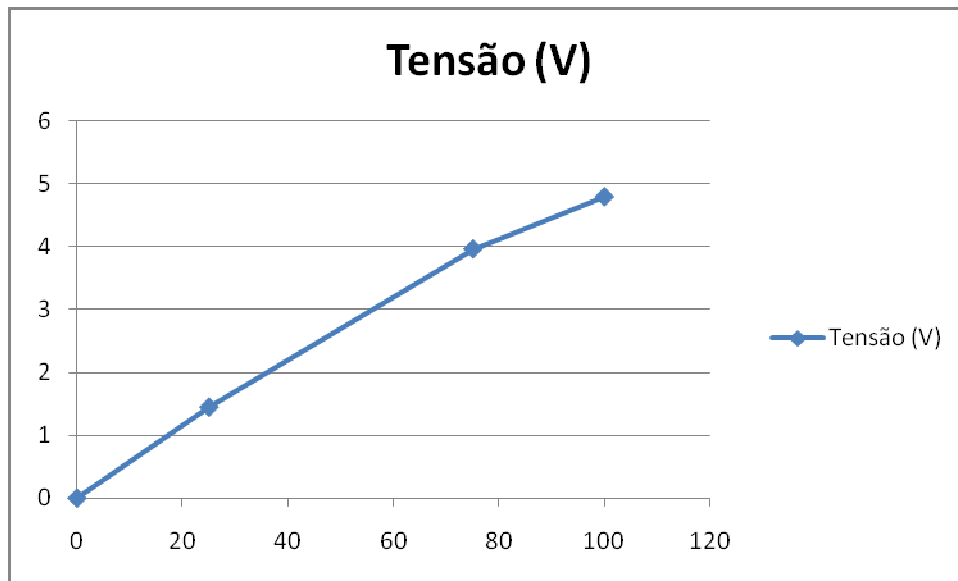


Figura 20 – percentagem de *Duty Cycle* x Tensão no pino do PIC

3.2.2.2 Implementação

Nesta seção, a programação do microcontrolador em linguagem Assembly apresenta suas principais etapas. Como o programa é bastante extenso e a programação não é intuitiva, apenas análises descritivas do mesmo serão feitas.

Anteriormente testamos o PWM para valores distintos e restritos para um duty cycle conhecido. Agora, uma abordagem mais complexa e abrangente deve tomar o seu lugar.

A seguir, um exemplo do código-fonte para o acionamento do controle por modulação PWM (1 e 2) é mostrado:

```

;-----
;   Função para setar pwm
;-----
setup_pwm

    clrf    PORTC           ; limpa a porta C
    bank_1           ; altera para banco 1
    movlw 0x00         ; move zero ao acumulador W
    movwf    TRISC       ; move o acumulador W para a porta C,
definindo-a como saída

    ;bcf    TRISC, 1 (zera TRISC,1)
    ;bcf    TRISC, 2 (zera TRISC,2)
    movlw 0xff         ; move 0ffH para o acumulador
    movlw    .127       ; valor máximo de tempo (numa escala
                        ; adimensional)
                        ; do pwm seria de 255 (decimal)
    movwf    PR2        ;* CONFIGURA PERIODO DO PWM
                        ;* T = (PR2+1).4.Tosc.TMR2_PRESALE
                        ;* T = 256.4.1us.16
                        ;* T = 1024*10^-6*16
    bank_0           ; altera para banco 0

```

```

movlw0x07          ; acumulador W recebe 07H
movwf      T2CON    ; T2CON recebe o valor de W
                ;bsf   T2CON, 1
                ;bsf   T2CON, 2
movlw0x0F          ;* CONFIGURA CCP1CON PARA MODO PWM
movwf      CCP1CON  ;* LIGA PWM 1
clrf      CCPR1L    ;* ZERA CCPR1L
movlw0x0F          ;* CONFIGURA CCP2CON PARA MODO PWM
movwf      CCP2CON  ;* LIGA PWM 2
clrf      CCPR2L    ;* ZERA CCPR2L

return           ; termina sub-rotina
;-----
;-----

```

A parte de configurar o PWM foi vista, mostrando seus principais registradores, variáveis e sequência dos mesmos dentro da sub-rotina.

Para a facilidade na etapa de programação, definimos no PIC16F877A:

Tabela 8- Definições práticas do programa principal

Flags de Configuração	• _WDT_OFF = watchdog timer desabilitado • _BODEN_OFF = brown-out reset desabilitado • _PWRTE_ON = Power-up Timer habilitado • _XT_OSC= oscilador cristal • _LVP_OFF = low voltage programing mode desabilitado • _DEBUG_OFF = in circuit debug desabilitado
Constantes do programa	• Baud Rate • Número de identificação dos robôs
Bits, Variáveis gerais e de interrupção	• Indicadores de Buffer • Start e Stop Bit

Com relação à comunicação serial, o recurso USART (Universal Synchronous Receiver Transmitter) foi usado no modo assíncrono, em que a comunicação é feita somente com duas vias; entretanto, como o modo não é sincronizado, essas duas vias são utilizadas para dados. Uma delas para transmissão (TX) e outra para recepção (RX). Isso possibilita que informações sejam enviadas e recebidas ao mesmo tempo (Full Duplex). A informação é recebida do computador através da porta paralela e serializada pelo microcontrolador.

A ‘sincronização’ do modo USART assíncrono é conseguida através do Baud Rate, que é a velocidade de transmissão. Essa transmissão trabalha com base nos bits (o Baud Rate geralmente é dado em bits por segundo) e assim conseguimos calcular o tempo de duração de cada bit. O sincronismo existente é feito para recepção de cada byte.

Formato do Caracter

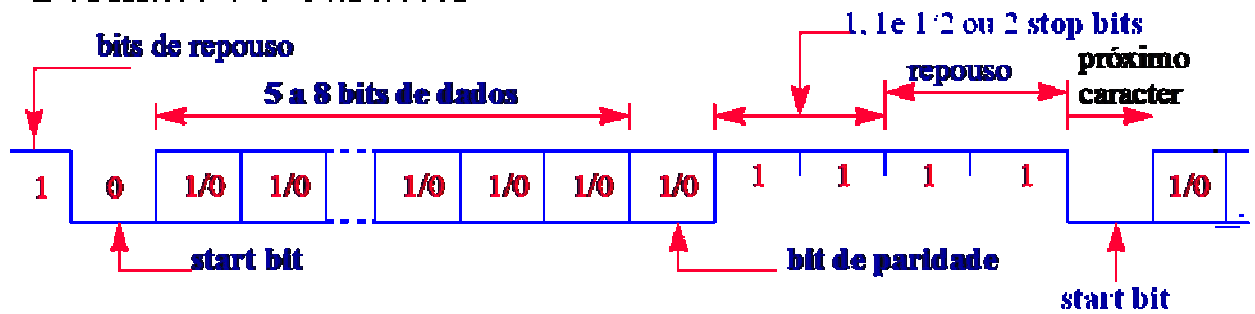


Figura 21– Comunicação serial assíncrona

As vias possuem estado padrão em nível lógico alto. A borda de descida do clock é reconhecida pelo RX e inicia o processo de sincronização para o start bit; esse pulso em nível baixo é chamado de Start Bit (ver Figura 21).

Depois de passar um tempo, RX tem o momento exato do Start Bit, deixa passar esse tempo pré-determinado e depois coleta 8 bits, pegando a sequência de dados que se encontra próximo do meio tempo do bit ($T_{BIT}/2$).

Quando TX envia um Stop Bit, o lado RX deve considerar a leitura do Stop Bit para que nenhum erro seja admitido durante a recepção. No projeto, definimos a Baud Rate em 9600kbps, o que nos dá $T_{BIT} = 104\mu s$.

A Figura 22 mostra resumidamente o fluxograma do programa principal que recebe via rádio, do sistema de estratégia, os vetores de velocidade que devem ser aplicados aos motores de 3 robôs, que são identificados pelo protocolo de comunicação. No tratamento dos números, a conversão de BCD para binário e vice-versa, de binário para código ASCII, de ASCII para ASCII2BCD e de ASCII para Hexadecimal foram incluídas em sub-rotinas do programa para que um padrão entre os dados recebidos, dados processados e dados de conversão convergissem ao mesmo sistema de unidades.

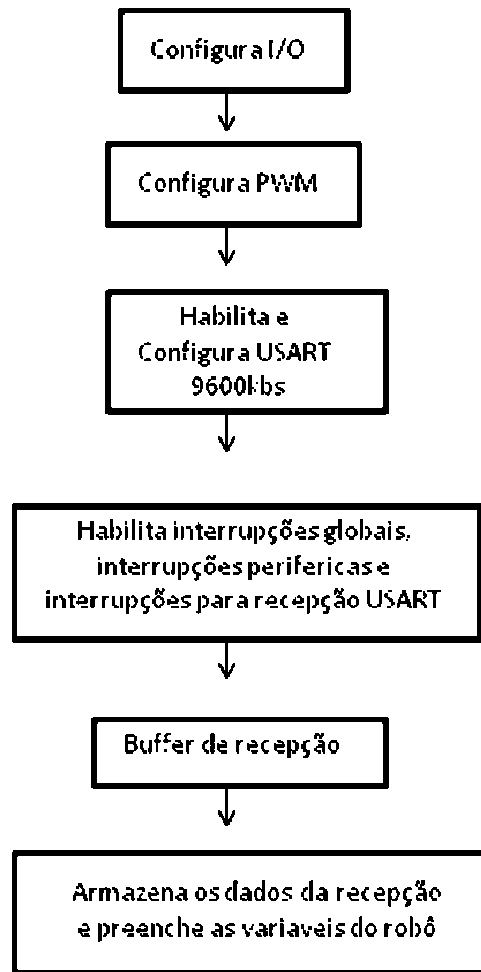


Figura 22– Fluxograma básico do programa principal

3.3 Motor DC (Corrente Contínua)

A vantagem de trabalharmos com um motor de corrente contínua é que neste tipo de dispositivo, obtemos uma quase perfeita linearidade entre a tensão aplicada em seus terminais e a velocidade angular do motor (Figura 23). Considerando a modelagem do sistema e as perdas, podemos conseguir essa relação, que é interessante no sentido que facilita os cálculos de movimento e portanto exige menos processamento. O primeiro relatório apresenta a linearidade

desta relação através da modelagem do motor: $w = \frac{k * V_a}{f * R_a}$.

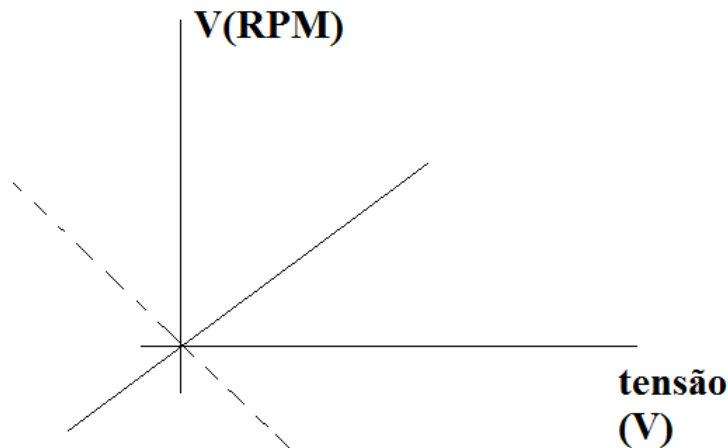


Figura 23 - Característica tensão x ω (velocidade angular)

O gráfico da Figura 23 representa a operação do motor DC em 4 quadrantes: 1 e 3: aceleração; 2 e 4: reversão ou frenagem. Essa linearidade justifica a aplicação comum dos motores DC em situações de controle de velocidade. Além disso, os motores de corrente contínua são uma alternativa mais atrativa em uma série de aplicações, já que são baratos e podem ser bem eficazes.

O motor utilizado nos robôs, pertence à categoria de ímã-permanente (baixa potência). A principal vantagem desse tipo de motor é que os ímãs não precisam de excitação externa nem dissipam a potência correspondente para criar campos magnéticos na máquina [4].

3.4 Sistema de controle

O controle desenvolvido mantém os robôs constantemente monitorados pela visão, que funciona como guia para a estratégia. Então, esta manda pulsos de PWM segundo orientação de velocidade calculada pelo sistema, alterando o duty-cycle, corrigindo a trajetória. Esse processamento da visão acontece em menos de 33 milissegundos, tempo suficiente para modificar a posição do robô com precisão, evidentemente desprezando as perdas do sistema mecânico e eletrônico em questão.

O sistema de controle de velocidade por posição ainda apresenta muitas deficiências e não funciona como esperávamos, uma vez que monitorados diretamente pela visão, erros são propagados sobre os sistemas de visão e estratégia, de forma que não se tem a precisão necessária para realizar os movimentos previstos.

A diferença de *duty cycles* do pulso de um motor para o outro faz o robô girar muito rápido quando, por exemplo, o robô realiza uma curva muito fechada. Devido ao peso do robô, relativamente alto, a sua inércia é grande, dificultando assim a mudança de percurso e controle sobre o mesmo.

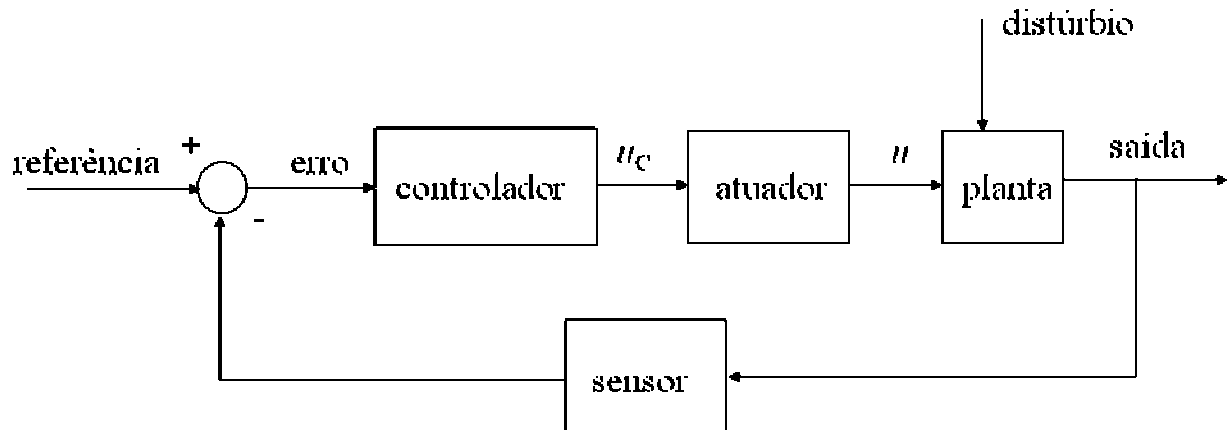


Figura 24– Estrutura básica de um sistema realimentado; u: ação de controle

Como mostra a Figura 24, o sistema de controle trabalha com um controlador PID por posição (x,y,θ) num sistema realimentado. O sistema tem uma posição de referência desejada, o controlador atua corrigindo o erro, calculado pelos dados do sensor, que no caso seria o sistema de visão.

O controle PID por posição possui a seguinte modelagem cinemática [13]:

$$\mathbf{v} = {}^v\mathbf{T}_\omega \cdot \omega_r \rightarrow \omega_r = {}^\omega\mathbf{T}_v \cdot \mathbf{v} \quad \Bigg|$$

onde :

$$\mathbf{v} = \begin{bmatrix} v \\ \omega \end{bmatrix} \quad \omega_r = \begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix} \quad {}^v\mathbf{T}_\omega = \begin{bmatrix} r_d/2 & r_e/2 \\ r_d/b & -r_e/b \end{bmatrix}$$

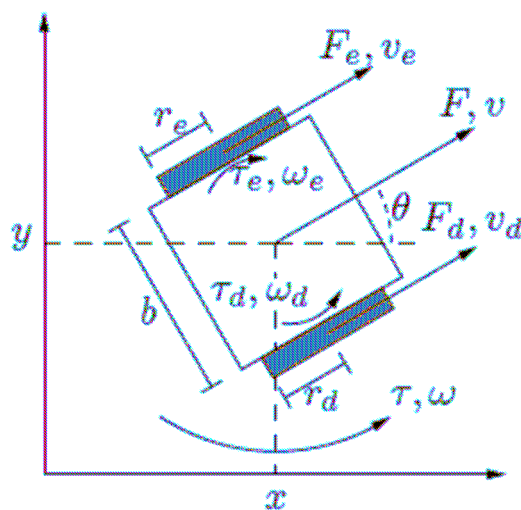


Figura 25 -Diagrama esquemático do robô

O robô modelado é sujeito a restrições não holonômicas, isto é, restrições nas possíveis velocidades que pode assumir, devido ao fato que o atrito impede que as rodas deslizem lateralmente. Assim, o robô só pode se movimentar na direção em que está orientado. Esta restrição é expressa matematicamente como:

$$\frac{dy}{dx} = \frac{\sin \theta}{\cos \theta} \Rightarrow \dot{y} \cos \theta - \dot{x} \sin \theta = 0$$

A relação entre velocidades no referencial do robô e velocidades em espaço de configuração é dada por:

$$\dot{\mathbf{q}} = {}^q\mathbf{T}_v \cdot \mathbf{v}$$

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad {}^q\mathbf{T}_v = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix}$$

3.5 Sistema Eletrônico

O PWM opera em períodos de tempo tão curtos que suas variações entre zero e máximo são imperceptíveis, mas a forma de onda equivalente possui voltagem menor. Dessa maneira é possível linearizar curvas através de um circuito digital. O PIC16F877 possui funções e chaves específicas para o tratamento de PWMs e estas serão usadas no decorrer do projeto.

A ponte-H serve para que inverter a polarização dos motores, sendo possível também inverter o sentido de giro dos mesmos. Esse mecanismo é baseado em transistores, sendo que dependendo dos pinos acionados (Alto - Baixo) de um microprocessador, a corrente fluirá ou não num certo sentido. Isto pode ser visualizado no esquema apresentado na Tabela abaixo.

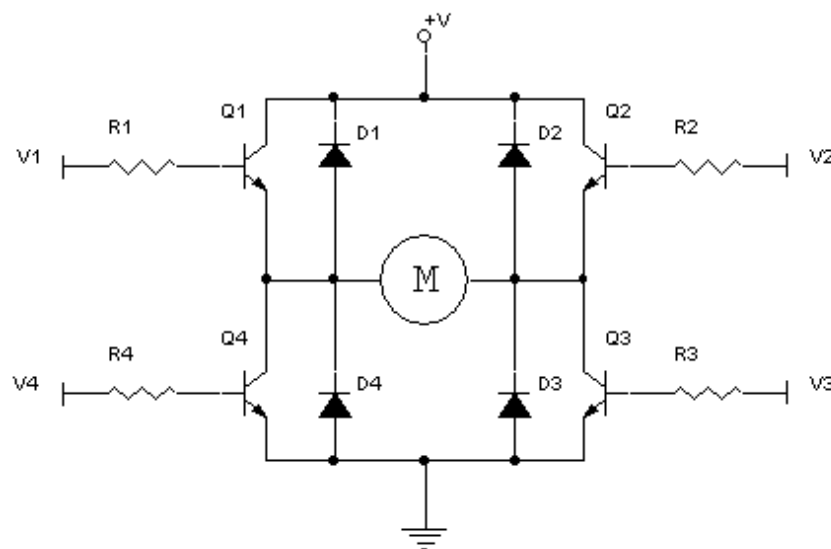


Figura 26- Circuito elétrico da Ponte -H

Tabela 9- Funcionamento da ponte-H.

V1	V2	V3	V4	Resultado
H	L	H	L	Q1 e Q3 fechados, corrente da esquerda para direita.
L	H	L	H	Q2, Q4 fechados, corrente da direita para a esquerda.
L	L	L	L	Motor desligado.

4. Conclusão

Este relatório apresentou detalhes de desenvolvimento do projeto, implementação e montagem do protótipo de um robô para a categoria 'very small size', a ser utilizado para participação em competições robóticas.

A parte mecânica está pronta: simples e robusta. Para efeito de pesquisa, a plataforma construída é suficiente para desenvolvermos o controle. Para efeito de competição, o projeto pode ser aperfeiçoado. A inclusão de encoders (sensoriamento interno) pode auxiliar as coordenadas do sistema de controle (no momento esse processo é realizado somente através da visão).

O projeto abordou um tema muito importante e de aplicação comum em muitos processos, tais como os motores que se deseja ter algum controle de velocidade, (independente da aplicação). Tratando-se de um sistema robótico, a construção de uma base mecânica é, de fato, fundamental para a implementação do projeto.

O projeto mecânico baseou-se em técnicas já utilizadas em máquinas comuns, como a utilização de rolamentos, a usinagem de peças e respectiva montagem de elementos, a redução do torque de motor por meio de engrenagens; a fim de obter uma eficiência melhor; otimizar o gasto de energia que para o futebol de robôs, assim como qualquer outra plataforma, é um fator crítico. Consumir menos energia implica em uma plataforma de alto desempenho, menos peso, maior robustez do carro do robô.

A parte de programação do microcontrolador, por trabalhar com a linguagem Assembly é complexa, devido aos cuidados em construir cada função e a configuração manual de registradores. Um programa simples de controle do motor possui mais de 300 linhas de código, evidenciando a falta de praticidade da linguagem.

Por outro lado, esse trabalho é compensador no sentido de que o programador adquire um maior cuidado com o espaço de memória, por exemplo; com a simplificação do código e a apresentação de comentários e identificação.

O PIC16F877A foi uma ferramenta essencial para a execução do projeto já que este possui o recurso PWM no módulo CPP, em que se pode modular uma faixa da largura de pulso, pré-determinado pelo oscilador.

Agradecimentos

Os autores agradecem a FAPESP e o CNPq pelo apoio recebido durante a realização do presente trabalho.

5. Bibliografia

1. PEREIRA, F. “*Microcontroladores PIC: Programação em C*”. São Paulo: Érica, 2005. 360 p.
2. FARIA, G.; ROMERO, R. A. F.; PRESTES, E.; IDIART, M. “*Comparing Harmonic Functions and Potential Fields in the Trajectory Control of Mobile Robots*”. In: IEEE RAM 2004 - IEEE Conf. on Robotics, Automation and Mechatronics, 2004, Singapura. Proceedings of IEEE RAM - Robotics, Automation and Mechatronics. <http://ieeexplore.ieee.org/sea> : IEEE, 2004. v. 2. p. 762-767.
3. Faria, G., “Uma arquitetura de controle Inteligente para Múltiplos Robôs”, Tese de Doutorado, ICMC, 92 p., 2006.
4. Grabbe, Eugene Munter “*Handbook of automation, computation, and control / prepared by a staff of specialists*”. Edited by Eugene M. Grabbe, Simon Ramo [and] Dean E. Wooldridge, 1958 - Wiley New York
5. Dorf, Richard C. “*Modern control systems*” / Richard C. Dorf 6th ed Reading, Mass. : Addison-Wesley, c1992
6. Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *IEEE International Conference on Robotics and Automation* (pp. 500-505). St. Louis, Missouri.
7. SOUZA, D. J.; LAVINIA, N. C. “*Conectando o PIC: Recursos Avançados*”. São Paulo: Érica, 2003. 384 p.
8. Santos, Davi Pereira dos; Penharbel, Éder Augusto; Faria, Gedson; Romero, Roseli Aparecida Francelin. Um sistema integrado para controle de um time de futebol de robôs.. p. 449-457. Congresso da SBC 26, Campo Grande, 2006, Workshop de Computação e Aplicações 1, Campo Grande, 2006. In: Anais. 0006. SEM CLASSIFICAÇÃO (evento com menos de 3 edições)
9. Bianchi, Reinaldo A. C.; Reali-Costa, Anna H. “O sistema de visão computacional do time FUTEPOLI de futebol de robôs”
10. Lopes, Alfredo, “Sci-Soccer – Desenvolvimento de uma Plataforma de Robôs Móveis voltada ao Futebol de Robôs”, FAPESP projeto PIPE 03/07973-2. (http://www.cientistasassociados.com.br/publicacoes/Enri_final.pdf)
11. "RoboCup - Robot World Cup Initiative", <http://www.robocup.org>.
12. Microchip 16 F87XA Datasheet 28/40/44-Pin Enhanced Flash Microcontrollers