

Processo Seletivo Phoenix 2016

Grupo 01

Participantes: Esdras R. Carmo – esdrasbrz@gmail.com – Computacional
Pedro Gabriel - pedrooculosgabriel@gmail.com – Elétrica

1 – Tipo de robô:

Depois de algumas pesquisas com o objetivo de encontrar um tipo de robô com características equilibradas de prós e contras, escolhemos o drumbot como base do projeto.

Dentre os motivos de escolha, podemos citar a alta robustez, a característica inversível que poderia ser aplicada, o centro de massa bem localizado, uma possível proteção das rodas pela armadura, arma resistente e alta colisão frontal.

As desvantagens da escolha desse modelo são o peso da arma muito elevado, sentido de impacto da arma muda quando invertido, necessário um motor exclusivo para arma e polia da arma possivelmente desprotegida.

O principal objetivo desse tipo de robô é, com o tambor, girar o robô adversário de modo a invertê-lo. O grande problema é que, contra robôs inversíveis, essa técnica se mostra ineficiente. Mesmo assim, a alta agilidade do robô, bem como sua eficiente armadura e reversibilidade faz com que ele seja altamente competitivo independente de seu adversário.

2 – Motores:

Para decidir quais motores seriam utilizados no robô primeiramente foi feita uma análise dos tipos de motores que podem ser utilizados, entre eles motor de passo, servo motor, motor DC, motor AC e a combustão interna.

O motor de passo é feito para se mover com muito mais precisão fazendo com que o torque por peso do motor seja menor do que a maioria, não sendo portanto uma boa escolha para esse tipo de aplicação.

Servo motor é feito a princípio para utilização considerando a posição angular do motor e é limitado a um limite de angulação, assim necessitando de modificações internas e no modo de controle para funcionar na movimentação das rodas.

Motores de corrente alternada (motor AC) são ótimos para esse tipo de aplicação atingindo boas taxas de velocidade e torque, porém são muito difíceis de alimentar utilizando baterias fazendo necessário uma eletrônica complexa para conseguir correntes negativas.

O motor de combustão interna tem um peso muito elevado em comparação aos outros tipos além de necessitar de tanque de combustível para seu funcionamento tornando-se assim inviável para essa utilização com peso limitado.

O motor de corrente contínua (motor DC) é fácil para se controlar e existem modelos que conseguem atingir uma boa taxa de torque por peso além de atingir velocidades de rotação relativamente altas tornando assim uma ótima escolha para esse tipo de aplicação, portanto será o modelo de motor que utilizaremos em nosso robô.

2.1 – Motores para movimentação

Em primeiro lugar é necessário saber qual será o peso em que esses motores terão de mover, assim para dimensionar o motor da melhor maneira utilizaremos o peso máximo do robô para os cálculos que será de 50 kg. Após isso veremos qual será o peso em cada roda para fazermos os cálculos assim:

$$\begin{aligned} 50\text{kg}/2 &= 25\text{kg} \\ 25\text{kg} \cdot 9,81\text{m/s}^2 &= 245,25\text{ N} \end{aligned}$$

Considerando a força efetiva em uma arena limpa com coeficiente de atrito de aproximadamente 0,9 temos:

$$245,25\text{N} \cdot 0,9 = 220,725\text{N}$$

Considerando a roda com 8cm de raio temos que o torque necessário será:

$$220,725\text{N} \cdot 0,08\text{m} = 17,658\text{Nm}$$

Para continuarmos temos que olhar algumas opções de motores que existem no mercado e uma boa maneira de encontrarmos um bom motor é olharmos para a P/peso e para I_{stall}/I_{no_load} que seria uma medida da corrente máxima que o motor suporta em relação a corrente consumida sem carga sobre o motor. Olhando essas informações encontramos um motor DeWalt de 24V com boas características sendo:

$$\begin{aligned} P/\text{peso} &= 2070,83 \\ I_{\text{stall}}/I_{\text{no_load}} &= 73,07 \end{aligned}$$

Sendo esses ótimos valores para um motor, agora podemos calcular se esse motor atingirá as especificações necessárias para o bom funcionamento do motor.

Primeiramente calculando o torque nominal do motor com base no coeficiente K_t em Nm/A, no caso de 0,01059, e na corrente nominal assim.

$$0,01059\text{Nm/A} \cdot 68\text{A} = 0,72012\text{Nm}$$

O valor é muito abaixo do necessário mas antes de decidir se não será possível usar temos de calcular a redução do sistema pois nenhum motor nessas proporções atingirá um torque tão alto, assim:

$$17,658\text{Nm} / 0,72012\text{Nm} = 24,52$$

Portanto o valor mínimo de redução deverá ser de 24,52 para 1. Entre algumas soluções comerciais temos uma caixa de redução com base em engrenagens planetárias de 25,91 para 1 assim utilizaremos essa caixa de redução para os cálculos.

$$0,72012\text{Nm} \cdot 25,91 = 18,658\text{Nm}$$

Portanto para o torque esse valor é mais que o suficiente para acelerar o robô, porém com uma caixa de redução de taxa tão grande temos que verificar se a rotação também será suficiente assim calculamos a rotação do motor pela fórmula, utilizando o fator K_v no caso de 880rpm/V e sendo um motor de 24V:

$$\begin{aligned} 880\text{rpm/V} \cdot 24\text{V} &= 21120\text{rpm} \\ 21120\text{rpm} / 25,91 &= 881,306\text{rpm} \end{aligned}$$

Esse valor é uma boa rotação para o sistema, agora calculamos a rotação real para o motor considerando a tensão fornecida menos a perda na resistência do motor, de 0,095 ΩHM.

$$\begin{aligned} 880\text{rpm/V} \cdot (24\text{V} - 0,095\Omega\text{HM} \cdot 68\text{A}) &= 15435,2\text{rpm} \\ (15435,2\text{rpm} \cdot 3,14 \cdot 2) / 60 &= 1615,55\text{rad/s} \end{aligned}$$

Agora verificaremos a velocidade linear para saber se atinge uma faixa boa para uma luta. Assim com a redução temos:

$$\begin{aligned} (1615,55\text{rad/s}) / 25,91 &= 62,35\text{rad/s} \\ 62,35\text{rad/s} \cdot 0,08\text{m} &= 4,988\text{m/s} \\ (4,988\text{m/s} \cdot 3600\text{s}) / 1000\text{m} &= 17,95\text{Km/h} \end{aligned}$$

Essa é uma boa velocidade para um robô de combate dessa categoria, portanto o motor pode ser usado para esse objetivo.

O motor escolhido é o DeWalt 24V.

Link para compra:

<http://www.robotmarketplace.com/products/BP389010-00.html>

3 – Baterias:

Para a escolha da bateria a ser utilizada no sistema de locomoção foram feitos alguns cálculos com base na tensão e na corrente necessária para alimentar os dois motores escolhidos assim temos que verificar primeiramente a necessidade, sendo que um piloto agressivo dirige acelerando em 50% da partida, a mesma dura 3 minutos ou seja 0,05h.

$$68A \cdot 0,05h \cdot 50\% = 1,7Ah$$

Para dois motores temos:

$$2 \cdot 1,7Ah = 3,4Ah$$

E a carga contínua consumida será de:

$$2 \cdot 68A = 136A$$

A bateria encontrada que melhor atinge essa faixa é uma bateria de Nicd de 24V com um fornecimento de 3 Ah e capacidade de fornecimento de 80A contínuo sendo assim:

$$2 \cdot 80A = 160A$$

Portanto será necessário alimentar os motores com dois conjuntos de baterias em paralelo fornecendo assim a corrente necessária para o funcionamento do motor e uma taxa de 6 Ah, assim também temos uma folga podendo alimentar também a eletrônica com o mesmo conjunto de baterias sem que exija muito do sistema, devido ao fato de que o consumo da eletrônica seja consideravelmente menor que o dos motores.

Devido ao fato de uma bateria de Nicd conseguir sustentar até um pico de duas vezes o valor calculado podemos ter um pico de:

$$2 \cdot 160A = 320A$$

Temos assim uma grande robustez do sistema.

A bateria escolhida é o modelo BPK-3000-24, cujo link é: <http://www.robotmarketplace.com/products/BPK-3000-24.html>

4 – Computacional:

Os problemas computacionais resolvidos nesse projeto foram o problema principal básico, que consiste em elaborar um algoritmo capaz de solucionar o problema do robô não conseguir andar perfeitamente em linha reta, mesmo enviando o mesmo sinal para os dois motores de movimentação e o case da visão, que trazia como problema a elaboração de um algoritmo que pudesse encontrar um cone em uma imagem pré-estabelecida e descobrir o ângulo de desvio entre o centro da imagem e o centro do retângulo que encobre totalmente o cone.

Para a resolução de cada um dos problemas, foi utilizada a linguagem Python e todos os códigos e rascunhos do projeto, incluindo uma versão para testar o problema computacional básico, está no seguinte repositório github: https://github.com/esdrasbrz/phoenix_ps

4.1 – Problema básico:

Para a resolução desse problema não foi utilizado nenhum sensor dos descritos na descrição do problema, apenas os métodos receberSinalControle(Lado I) e rotacionarMotores(Lado I, int sinal). Além disso, foi suposto que o RPM máximo atingido por cada motor era de 15435,2, adequando-se assim ao motor utilizado no projeto.

Para elaboração do algoritmo, foi suposto que o sinal positivo renderia a rotação máxima do motor, isto é, quando o motor recebe sinal 100 ele rotaciona mais do que com sinal -100 em sentido oposto. Assim, foi deduzido que quando o motor direito rotaciona recebe sinal positivo e o esquerdo negativo, de mesmo módulo do primeiro, o robô estaria andando para frente e tendendo para a esquerda. O contrário também é válido, quando o esquerdo recebe sinal positivo e o direito negativo de mesmo módulo, o robô anda para trás tendendo à direita.

Para tratar então do problema, foi elaborado um método chamado `rpmReal(sinal)` dentro do arquivo `prob_computacional.py` que faz a leitura do sinal em inteiro no intervalo $(0, 100]$ e, com base na tabela dada junto ao problema, retorna o RPM real que deveria ser aplicado no motor que rotaciona mais a fim de compensar a perda do outro motor que rotaciona menos. A razão para retirar de um motor que rotaciona mais em vez de acrescentar no motor que rotaciona menos é simples: caso os motores recebam sinal 100, o que rotaciona menos não teria como rotacionar mais a ponto de compensar a diferença, já que se encontra no máximo de rotação possível para aquele sentido.

Os problemas que poderiam ocorrer com esse algoritmo é com base nos poucos dados da tabela. Dessa forma, caso o robô recebesse sinal 9 e -9 para se locomover, o algoritmo não faria nenhuma correção pois entraria no caso $(0, 10)$ em que a diferença em porcentagem é 0. No entanto, na prática isso provavelmente não aconteceria, já que o desvio de rpm de cada motor deve seguir um padrão de acréscimo com base no sinal enviado, então o robô se locomoveria levemente desviado da linha reta nesses casos específicos.

4.2 – Case da visão:

Nesse problema, foi utilizado os comandos dados com o problema blob e threshold. Dessa forma, foi fácil retornar as coordenadas dos retângulos que encobrem totalmente as áreas laranja da imagem.

O grande problema, então, foi o fato de tanto o cone quanto os garis serem laranja na imagem, não tendo como distingui-los apenas pela cor. Deste modo, foi necessário a compreensão que os garis, como estão sempre deitados (haja vista os pontos que deveria assumir na resolução do problema), possuem sempre o comprimento da base maior que o comprimento da altura. O cone, no entanto, possui exatamente o contrário: comprimento da altura maior que o da base. Assim, com um simples condicional, foi resolvido o problema para distinguir entre o cone e os garis.

O segundo desafio deste case foi encontrar o ângulo entre o centro da imagem e o centro do retângulo que cobre totalmente o cone. Para isso, foi utilizado a biblioteca `math` do Python, que contém o método `atan(tan)` que retorna o ângulo em radianos que corresponde à tangente dada na entrada (`tan`). Assim, com as coordenadas do centro da imagem, basta uma simples trigonometria para descobrir a tangente do ângulo, que corresponde à $(\text{cone_y} - \text{centro_y}) / (\text{cone_x} - \text{centro_x})$.

Os problemas desse algoritmo diz respeito a casos em que a imagem não se adequa perfeitamente ao que foi enunciado no problema, como o gari em pé ou um cone cuja base é maior que a altura.