**Problemas Computacionais**

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

**Problema básico:**

Para a resolução desse problema não foi utilizado nenhum sensor dos descritos na descrição do problema, apenas os métodos receberSinalControle(Lado l) e rotacionarMotores(Lado l, 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.

**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 (cone\_y - centro\_y) / (cone\_x – 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.