

Fig. 1.b - Representação da condição final.

O ambiente foi montado utilizando uma base branca que foi dividida em uma matriz 10x10, essa divisão foi feita com fita isolante fosca preservando um espaço interno em cada unidade de  $20\text{ cm}^2$ , a espessura desta fita é de  $19\text{ mm}$ . A área total do ambiente é de  $2,2\text{ m}^2$ .

Como pode ser visualizado nas figuras 1.a e 1.b as áreas pintadas de amarelo nas extremidades representam as posições iniciais dos robôs e onde o tubo preto supostamente danificado deve ficar ao final da missão; Os tubos azuis espalhados pelo ambiente serviram para a construção do novo caminho; O bloco laranja com uma curva em  $90^\circ$  pintada na sua superfície superior na cor azul é móvel e também fará parte do novo caminho; Os canos e bloco verdes são fixos e fazem parte da estrutura já existente; Os blocos vermelhos com setas pintadas de amarelo na sua superfície superior são fixos e em dois deles existem pinos que representam válvulas.

#### IV. PROCEDIMENTOS

Foram construídos vários protótipos para a solução do desafio, onde foi escolhido apenas um modelo para a montagem dos dois robôs que atuarão na competição, a escolha foi baseada em testes onde o modelo demonstrou ser mais eficiente na realização da tarefa proposta, atendendo também as limitações propostas pelo desafio. A seguir será descrito o projeto de sua estrutura, locomoção e solução apresentada para o restabelecimento da produção de petróleo.

##### A. Principais desafios encontrados no desenvolvimento dos robôs

As principais dificuldades encontradas para a construção do modelo final foram enquadrar o robô dentro da especificação de tamanho e seguir a linha com movimentação suave de ajuste.

##### B. Modelo escolhido

O modelo atual escolhido apresenta as dimensões de  $22 \times 18 \times 20\text{ cm}$ , utiliza três atuadores, quatro sensores e um controlador (“brick”). Optamos por um robô compacto pelo fato do cenário do desafio apresentar vários tubos espalhados e a necessidade de movimentação entre eles e próximo a parede.

Para o sistema de locomoção foram utilizadas duas rodas com tração independente e uma roda rodízio situada atrás.

Durante o desafio os robôs terão que realizar tarefas para o conserto dutoviário como: levantar, carregar e descer tubos; empurrar blocos que representam curva de tubos e fechar e abrir a válvula da plataforma.

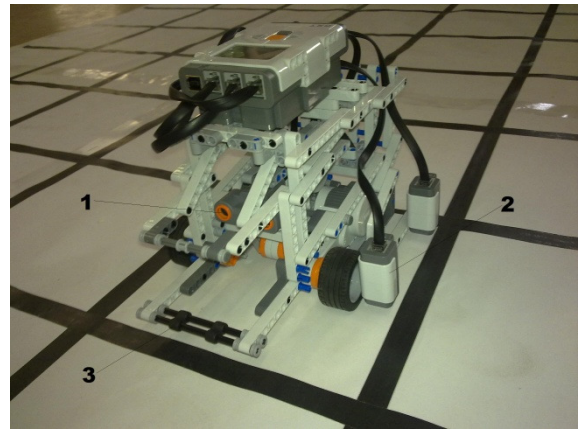


Fig. 2. - Modelo atual dos robôs (visão frontal em perspectiva)

A Figura 2 apresenta o modelo atual dos Robôs Spiritus e Corpus com um apontamento de alguns dos dispositivos utilizados na sua estrutura especificando o funcionamento de cada um:

- 1) Sensor sonar responsável por detectar os tubos e anteparos;
- 2) Sensores de luz responsáveis pela navegação orientada pelas linhas;
- 3) Sistema para movimentação dos tubos que é acionado por um motor e apresenta um apoio móvel situado na parte superior para evitar que o tubo escorregue.

A figura 3 mostra outra perspectiva do robô apontando outros componentes da estrutura:

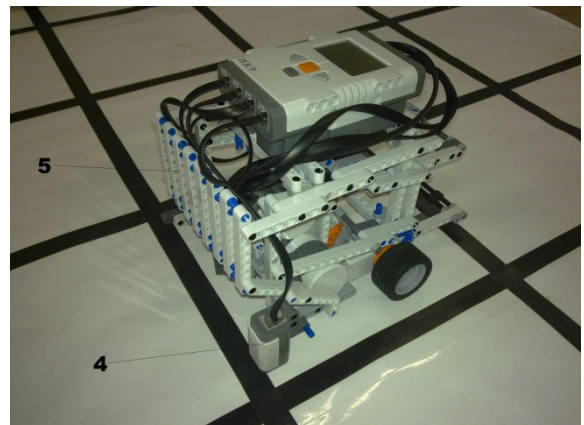


Fig. 3. - Modelo atual dos robôs (visão posterior em perspectiva)

- 4) Uso de sensor de luz, situado em um dos lados do robô, responsável pela outra parte da navegação, a contagem de linhas;
- 5) Uma parede na parte de trás do robô foi montada para posicionar os blocos e o acionamento das válvulas.

Portanto, como resultado dos testes feitos com os protótipos montados, o modelo adotado é bastante simples e pequeno, dentro das especificações da competição, e consegue atuar de maneira satisfatória para o desafio proposto.

### C. Utilização do JCreator

O JCreator é um software, utilizado para desenvolvimento Java e Web, oferecido em freeware e shareware. Para o desenvolvimento dos programas dos robôs, foi utilizada a versão freeware, instalando nesta, as bibliotecas de classes do leJOS NXJ 0.85 – a API de desenvolvimento Java [2] para o Lego NXT Mindstorms 2.0, bem como o compilador para a API.

A facilidade de implementação, manutenção, simplicidade de código, bem como a possibilidade de se implementar Programação Orientada a Objetos, permitindo assim dividir as tarefas em métodos, foram fatores primordiais para a escolha do Java, como linguagem de programação, bem como o leJOS 0.85, por ser uma API “open-source”. Esta escolha foi muito bem aceita pela equipe, de forma que o programa de cada robô pudesse ser dividido por toda a equipe. Dessa forma, nenhum integrante fez o mesmo programa; todos escreveram programas diferentes, que continham as rotinas (métodos) que realizam as tarefas do robô, como levantar garra, girar, acelerar, desacelerar, etc.

Muitos testes foram realizados para estudar a API e a sua implementação nos robôs. Calibração, erros de medição, e entre outros quesitos foram avaliados nos testes. A partir destes, foram desenvolvidas as rotinas (métodos) de funcionamento do robô. Nessa fase, reduziu-se a programação do robô em ações (levantar garra, translação, rotação, aceleração, desaceleração, etc.), reunindo mais tarde, todas as rotinas prontas em um único método – o método “main” – do programa principal de cada robô.

Para o ambiente definido numa matriz de espaços, o controle do posicionamento dos robôs é feito utilizando-se sensores de luz que contam as linhas pelas quais os robôs passam. Caso os robôs atinjam determinadas posições na matriz, rotinas são ativadas [3]:

1) *Pegar e mover os canos*: em vários momentos durante a execução do programa o robô terá que mover os canos do lugar. Quando o robô estiver próximo deles, ou seja, quando atingirem determinados pontos da matriz, a rotina será ativada. Utilizando um sensor de ultra-som, o robô perceberá qual o momento certo de levantar o cano e depois de andar até a posição definida pela programação, ele soltará o cano.

2) *Liberar e travar o fluxo de petróleo dos tubos*: por duas vezes os robôs precisam ativar travas que representam a liberação ou o travamento das plataformas de petróleo. Para executar essa tarefa eles utilizam uma estrutura posicionada na parte traseira do corpo do robô. Esse processo é realizado no início do programa e ao final da construção do duto alternativo.

3) *Gerenciamento do posicionamento dos robôs em relação ao ambiente*: Para evitar que os robôs percam sua noção de localização dentro da arena foi implementado um algoritmo de controle que utiliza sensores de cor para ajustar o posicionamento dos robôs entre as linhas da arena.

### D. Comunicação

Como parte fundamental da solução do desafio proposto, existe a necessidade dos robôs interagirem entre si, para assim completar a missão de forma cooperativa e dinâmica. A partir disto foi analisada a melhor forma de utilizar uma malha de comunicação, onde se resolveu que os dois robôs a partir de um determinado ponto da arena passem a informar um ao outro suas posições, sentido da movimentação, última ação realizada e a meta seguinte. Com isso ambos têm a capacidade de não se colocar em rota que atrapalhe o próximo objetivo do companheiro, e ainda recebem uma realimentação das atividades realizadas, proporcionando, dependendo da situação, permissividade na realização de ações seguintes. Para efetivar essa proposta foi utilizada a comunicação padrão *bluetooth* [4] entre os robôs, pois esta oferece a versatilidade das redes *wireless*, além de ser um recurso nativo do kit lego Mindstorms NXT 2.0, o qual foi utilizado como base para resolução do desafio.

## V. CONCLUSÃO

Este artigo tratou do desafio proposto na categoria IEEE Standard Educacional Kits da Competição Brasileira de Robótica 2010, demonstrando as técnicas e soluções adotadas pela equipe GERAIE no seu cumprimento. As dificuldades foram inúmeras, porém foram as grandes responsáveis pelo desenvolvimento e aperfeiçoamento da equipe. Os estudos e discussões constantes a cerca das técnicas possibilitaram a evolução e aprimoramento das nossas soluções.

## VI. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Regras da LARC 2010, categoria IEEE SEK, Disponível: <http://www.cbr10.fei.edu.br/IEEESEK.htm>
- [2] API do leJOS. Disponível: <http://lejos.sourceforge.net/nxt/nxj/api/index.html>
- [3] S. Roland, N. Illah, *Introduction to Autonomous Mobile Robots*. The MIT Press, Cambridge, 2004.
- [4] NXT Bluetooth and Control, Disponível: <http://mindstormsnxt.blogspot.com/2006/08/nxt-bluetooth-and-control.html>