

COOPERAÇÃO DE ROBÔS MÓVEIS TERRESTRE E AQUÁTICO PARA O TRANSPORTE DE CARGAS

RAFAEL LUCAS CAMARGOS*, GUILHERME GAZZINELLI ROHRMANN*, RENATO DE SOUSA DÂMASO*

*GER, Laboratório de Robótica, Departamento de Engenharia Mecatrônica
CEFET-MG Divinópolis, MG, Brasil

Emails: rafaellucascamargos@gmail.com, guilherme.gazzinelli@gmail.com,
renatosd@div.cefetmg.br

Abstract— This paper presents the design of a robotic system to solve the Open category Latin American Robotics Competition (LARC) challenge in 2015. It consists in the transport of barrels in an aquatic scenario simulating a port and an oil platform. For this, was developed two robots that work cooperatively. The first was a mobile robot crawler, called the forklift, and an aquatic robot ferry type. The forklift had odometry sensors and a computer vision, implemented through a smartphone with Android operating system. This element had a function to go through the port and oil platform in order to locate, collect and transport the barrels, boarding them on the ferry. The ferry was equipped with an Arduino microcontroller for the control.

Keywords— Autonomous Mobile Robots, Cooperative Robotics, Mobile Robot Navigation

Resumo— O presente artigo apresenta o projeto de um sistema robótico desenvolvido a partir de um desafio proposto para a categoria Open da Competição Latino-americana de Robótica (LARC) em 2015. Este propunha o transporte de barris em um cenário simulando um porto e uma plataforma de petróleo. Foram assim desenvolvidos dois robôs que atuam de forma cooperativa, sendo um robô móvel a esteiras, chamado de Empilhadeira, e um robô aquático do tipo balsa. A Empilhadeira percorre o porto e a plataforma de petróleo no intuito de localizar, recolher e transportar os barris, embarcando-os na Balsa. Já a Balsa possui um controle baseado no microcontrolador Arduino, em constante contato com o *smartphone*.

Palavras-chave— Robôs Móveis Autônomos, Robótica Cooperativa, Navegação de Robôs Móveis

1 Introdução

A robótica cooperativa vem sendo discutida academicamente por pelo menos mais de duas décadas como observado em (Ozaki et al., 1993), (Asama et al., 1996) e (Alami, 1996). Várias formas de cooperação são estudadas, mas a base principal constitui-se na ajuda mútua de robôs, quando a tarefa não pode ou é ineficaz se ser realizada de forma individual. Essas limitações podem ser derivadas dos ambientes terrestre, aéreo ou aquático, ou de acordo com a finalidade do robô (transporte, mapeamento). O transporte autônomo de cargas é um assunto que também será abordado neste artigo, bem como pode ser visto em (Stilwell and Bay, 1993), que discute o robô autônomo para esse mesmo objetivo. Nesta mesma vertente, estratégias inteligentes são desenvolvidas para navegação de robôs autônomos isolados ou em grupo, como visto em (Pessin, 2013).

Este artigo é baseado na solução do desafio proposto pelo Concurso Latino Americano de Robótica - LARC 2015, cujo cenário é apresentado na Figura 1, fundamentado em uma aplicação real de um ambiente marítimo. Para isso, tem-se um tanque com dimensões de 400x200 cm e 70 cm de altura, onde é simulado um porto e uma estação petrolífera. Nesta simulação, o porto contém suprimentos para a plataforma de petróleo, cubos amarelos, e do lado da plataforma flutuante estão os barris de petróleo, cubos vermelhos. Os barris são cubos de 50mm de aresta feitos de ABS.

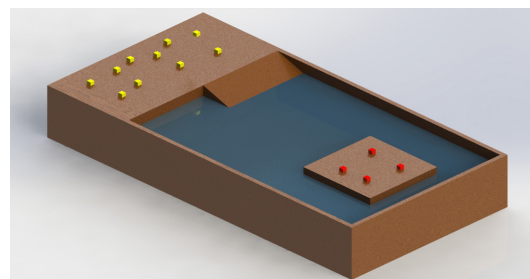


Figura 1: Cenário proposto para o transporte de cargas

O objetivo principal é levar os suprimentos para a plataforma e buscar os barris para o porto de forma autônoma. O problema em si sugere várias soluções possíveis e o trabalho desenvolvido abordou uma delas, a de usar um robô terrestre para pegar os objetos no campo e outro aquático, para transportar o robô terrestre com sua carga. Visto que cada um possui suas limitações e alcance num determinado campo de trabalho, é necessário que os dois se comuniquem e atuem de forma cooperativa.

2 Os Robôs

Cada um dos robôs desenvolvidos, possui sua função específica e comunicam entre si ao longo de toda a execução da tarefa para que, de forma cooperativa, consigam efetuar o transporte das cargas. A Empilhadeira (robô terrestre) tem a função de coletar e transportar a carga e a Balsa (robô

aquático) tem a função de transportar o robô terrestre. O fluxo de dados entre os robôs é representado por um diagrama na Figura 2.

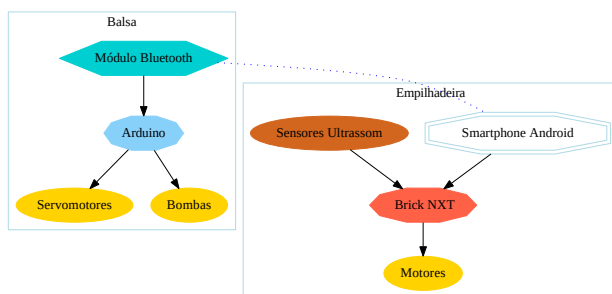


Figura 2: Diagrama de fluxo de informações nos robôs

Como pode ser visto através do diagrama, o robô Empilhadeira é movimentado controlando os motores através das instruções do *Smartphone* que são enviadas ao Brick NXT, que indica para onde virar, enquanto busca os suprimentos/barris, quando e como voltar para o robô Balsa, ao passo que, o restante da movimentação é calculada a partir de odometria.

Enquanto isso é realizado, o robô Balsa aguarda o comando da Empilhadeira. Assim são recolhidas as rampas de passagem dando início à movimentação para a travessia aquática realizando o sensoramento do ambiente e controle dos atuadores.

2.1 Empilhadeira

A Empilhadeira foi desenvolvida usando peças de kits Lego Mindstorms NXT 2.0. A montagem mecânica foi feita para que o robô possuísse uma constituição diferencial e tração por esteiras na movimentação, dando uma maior fricção mesmo em terrenos mais irregulares, permitindo vencer inclinações que poderiam apresentar-se variáveis durante a realização da tarefa. Isso inclui também a funcionalidade de fazer com que o robô não fique preso em pequenas irregularidades, como as entradas das rampas de embarque e desembarque da Empilhadeira na Balsa, e durante a movimentação, que gera uma inclinação da plataforma de petróleo em função de sua baixa fluatibilidade.

Foi necessário ainda, incluir um terceiro motor junto de um sensor de contato e mais esteiras, para realizar o levantamento do suprimento/barril, trazendo para o robô com segurança para que este não corresse o risco de ficar preso em alguma superfície inclinada ou rugosa, o sensor de contato identifica o fim do movimento de ascensão do objeto. A Figura 3 mostra a montagem física do robô Empilhadeira.

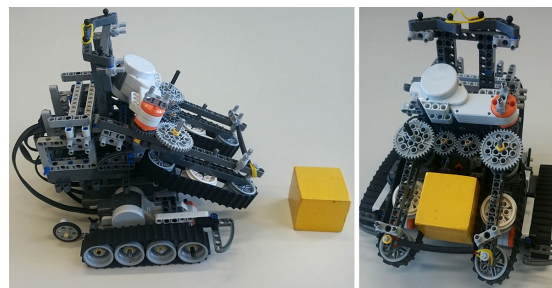


Figura 3: Montagem física da Empilhadeira próxima a/carregando um bloco amarelo de suprimento

Além da montagem física usando as peças de kit Lego, também foi incluído no robô um celular *Smartphone* com sistema operacional *Android*. Este possui um processador de núcleo quádruplo 2.5 GHz, com 2 Gb de memória RAM e câmera de 16 Mp.

2.2 Balsa

A balsa é um tipo de veículo de transporte aquático frequentemente utilizado em águas rasas. Possui grande capacidade de carga e estabilidade, o que conduziu ao uso deste modelo como fonte de inspiração para o robô.

O projeto do robô Balsa foi dividido em mecânico, eletromecânico e eletrônico. A parte mecânica foi desenvolvida utilizando um sistema CAD, visto na Figura 4a e Figura 4b e consiste no sistema de flutuação, rampas, assoalho e paredes. O flutuador utiliza como matéria prima o isopor, enquanto o restante dos componentes mecânicos foram construídos em MDF, através de corte a laser e alguns elementos realizados por meio de sistema de prototipagem rápida. Estes foram impermeabilizados por meio de aplicação de verniz, cola e pintura.

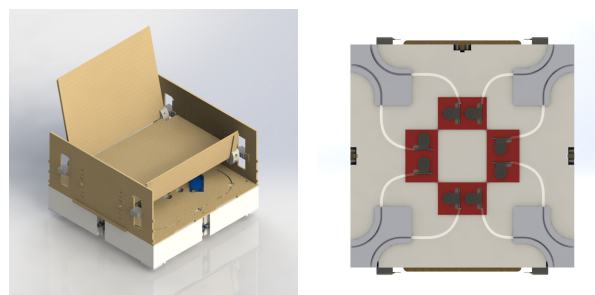


Figura 4a: Vista de perspectiva do robô Balsa
Figura 4b: Vista inferior do robô Balsa

No sistema eletromecânico estão os servos, para controle dos ângulos das rampas, e motores de propulsão (eletrobombas). Os servomotores são afixados nas paredes laterais, consistindo de 4 motores, um a cada lado das rampas. Estes são responsáveis pela alteração do ângulo que permite com que a Empilhadeira embarque e desembarque.

Na parte inferior estão os elementos propulsores, que se trata de um conjunto formado por 8 eletrobombas que compõem a capacidade de deslocamento omnidirecional. O seu arranjo foi elaborado como é mostrado na Figura 4b, sendo agrupadas em quatro grupos de duas, cada um destes responsável por locomover em um sentido. Estes motores trabalham de forma independente, muitas vezes trabalhando de forma diferencial. Cada bomba faz a admissão da água através de um duto que se mantém sob a superfície e a bombeia por meio de mangueiras para a parte lateral do veículo. O sistema se assemelha ao arranjo utilizado por *Jet Ski*, porém neste caso com mais propulsores que garantem a flexibilidade na locomoção. A lógica de locomoção fornece quais bombas devem ser acionadas para a realização do movimento desejado. Para deslocar-se para a frente as bombas situadas na parte de trás são acionadas. Desta forma, para locomover-se em um determinado sentido, os motores do lado oposto ao movimento devem ser acionados. O arranjo ainda permite rotações em torno do próprio eixo, com uma lógica diferencial que será melhor definida posteriormente neste artigo. Quatro bombas são acionadas de forma intercalada para realizar a rotação, de acordo com a direção que deseja-se rotacionar.

A eletrônica é implementada em um sistema microprocessado baseado na placa de desenvolvimento Arduino MEGA, com processador de 16 MHz. Conta com um módulo de comunicação *bluetooth*, responsável pela recepção e envio de comandos para o dispositivo de comando da Empilhadeira. Há também eletrônica de potência embarcada, que é responsável pelo acionamento das bombas, regulação de tensão para os servomotores.

O sensoriamento é realizado por meio de 5 sensores de ultrassom posicionados de forma a possibilitar leitura das distâncias e assim obter a localização no campo de competição. Utilizando dois sensores do lado direito permite que se conheça a diferença entre a distância entre a parte frontal da Balsa e a traseira, pela qual é possível manter a Balsa locomovendo-se paralelamente à lateral. Garantir estas medidas é de suma importância para que o sistema de processamento conheça sua localização no ambiente.

3 Fusão de odometria e visão computacional na movimentação do robô terrestre

O kit Lego Mindstorms NXT 2.0 possui várias aplicabilidades dentro da engenharia, principalmente como uma ferramenta de ensino, como visto em (Lew et al., 2010). A programação da Empilhadeira foi feita em linguagem Java, através do firmware LeJOS, que permite o uso de uma programação orientada a objetos (Java), *Threads*, re-

cursão, sincronização e outros recursos com uma API bem documentada (*LeJOS, Java for Lego Mindstorms*, n.d.) e uma biblioteca consistente com várias classes *open-source* já criadas e disponíveis.

Usando algumas das bibliotecas disponíveis em conjunto com o firmware LeJOS foram desenvolvidos os algoritmos de movimentação da empilhadeira, das quais foram utilizadas, principalmente, classes para a odometria. Porém, somente esta não é suficiente para a execução da tarefa, visto que devido a erros aleatórios provenientes do escorregamento da correia de transmissão e outros relativos à movimentação acabam por gerar incertezas na estimação do posicionamento. Por isso, também foi usada a biblioteca OpenCV embarcada no Smartphone Android, executando funções de processamento gráfico e localização dos elementos principais na imagem, assim como suas respectivas áreas. Usando essa biblioteca, o celular Android pôde realizar a captura e o processamento das imagens, tomada de decisões, e por fim, a comunicação com a parte de movimentação da Empilhadeira, através do Brick NXT e com robô aquático, através do módulo Bluetooth.

O algoritmo elaborado faz com que a Empilhadeira varie seu comportamento alternando entre o uso da visão computacional e a odometria para se movimentar, enquanto busca pelos barris, retorna para a Balsa, segue a referência para estacionar dentro da Balsa e deposita os barris pegos.

Na busca pelos barris, a odometria é usada de forma a ter o melhor campo de visão no terreno, enquanto usa esse método para se locomover, o algoritmo de visão computacional busca por objetos que correspondam ao procurado, filtrando através da cor, fazendo a segmentação e separação dos objetos. Desta forma, assim que é encontrando um objeto correspondente ao procurado, a Empilhadeira alterna sua movimentação, deixando de lado o seguimento dos pontos matriciais e passando a seguir o objeto através da visão computacional.

Para isso, a imagem é primeiramente desfocada e tem sua resolução diminuída, em seguida convertida para HSV. Com a imagem em HSV, é feita a seleção dos *pixels* de interesse através de um filtro realizado com mínimos e máximos para os três parâmetros desse tipo de representação gráfica, ($H_{min.}$, $H_{max.}$, $S_{min.}$, $S_{max.}$, $V_{min.}$ e $V_{max.}$) e, por fim, essa imagem é dilatada, para eliminar ruídos em volta dos elementos. Na Figura 5 é mostrada a imagem original ao lado do resultado após feita a seleção e aglomeração dos pixels de interesse e por último a imagem do aplicativo sendo executado no smartphone, com os objetos de interesse contornados.

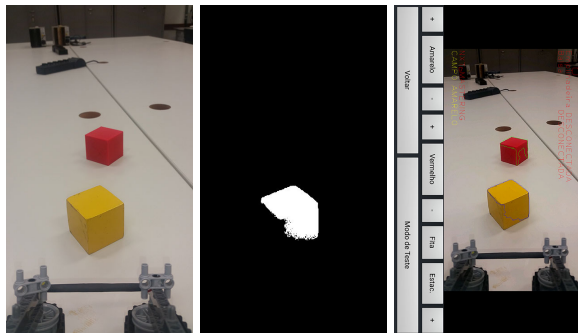


Figura 5: Etapas da visão computacional na Empilhadeira

Com os *pixels* selecionados e aglomerados, são encontrados os respectivos centros de massa e áreas para cada elemento de interesse, com isso, é feita uma organização destes em um vetor, cada elemento possuindo o centro de massa em x , y e a área A daquele objeto. Este procedimento é realizado para cada tipo de barril e marcação utilizada na Balsa, criando n vetores de tamanho m desse tipo, para cada tipo de objeto no cenário que necessitam de detecção através da visão.

Calculadas as áreas e centros de massa de cada objeto desejado, a Empilhadeira passa a executar um outro algoritmo para busca, coleta dos barris e volta para a Balsa. Considerando que o vetor de barris desejados possua tamanho n , seja x_c o valor do centro de massa em x do maior barril, y_c , em y , L_w a largura total da tela do Smartphone em pixels, e L_h a altura total da tela do Smartphone em pixels, onde a tela é dividida em quatro regiões: a primeira compreende tudo que estiver de 0 a $0,25 \times L_h$ (parte inferior da tela) a segunda, 0 a $0,4 \times L_w$ (parte esquerda), a terceira, $0,4 \times L_w$ a $0,6 \times L_w$ (meio) e a última, $0,6 \times L_w$ a L_w . Dividindo a tela nessas quatro áreas, caso o objeto desejado esteja na primeira, a Empilhadeira executa a função de coletá-lo, caso na segunda, gira para a esquerda, caso na terceira, move-se para frente, e caso esteja na última, gira para a direita, conforme descrito também no Algoritmo 1.

Além dessa finalidade, é usada a visão computacional para reiniciar os valores da odometria através de uma listra roxa colocada na Balsa, anulando assim os erros acumulados. Isto é feito através de marcações verdes na Balsa, que são detectadas quando a Empilhadeira entra na Balsa, e através de um valor ajustado mínimo para o centro de massa, esta identifica que foi atingido o meio da Balsa, com isso é reiniciado o valor acumulado da pose provida pelo sistema de odometria para o eixo y . Enquanto retorna para a Balsa é usado o mesmo algoritmo de forma semelhante, modificando o objeto no qual se procura e a condição de parada.

enquanto $m = 0$ faça

Movimenta através da odometria

fim enquanto

se $n > 0$ então

se $y_c < 0,25 L_h$ então

Coleta o objeto e volta para a Balsa

senão

se $x_c > 0,4 \times L_w$ e $x_c < 0,6 \times L_w$ então

Segue em frente

senão

se $x_c > 0$ e $x_c < 0,4 \times L_w$ então

Gira para a esquerda

senão

Gira para a direita

fim se

fim se

fim se

fim se

4 Estratégia de transporte aquático autônomo

A Balsa sendo responsável pelo transporte aquático, deve aguardar atracada ao porto, com a rampa traseira abaixada até que a Empilhadeira esteja devidamente posicionada e envie o comando permitindo que se inicie o processo de transporte. Após o recebimento da Empilhadeira, o robô Balsa eleva a rampa traseira e inicia o processo de deslocamento até o porto. Neste momento são iniciados dois procedimentos simultâneos, o controle de angulação, para manter o paralelismo com a borda do tanque e o controle de deslocamento. O controle de deslocamento utiliza os sensores traseiro e frontal para localizar-se no sentido y enquanto os sensores laterais fornecem a posição em x , método este, sintetizado na Figura 6.

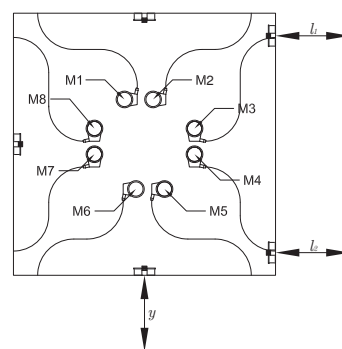


Figura 6: Configuração dos atuadores/sensores da Balsa e suas principais variáveis

Após o posicionamento ser efetuado é realizado o procedimento de atracagem na plataforma, onde os motores são acionados em alta potência, a rampa frontal abaixa e logo um comando é enviado à Empilhadeira para que ela desça. O barco fica atracado desta forma até que a Empilhadeira retorne e envie o comando de desatracar, com a

subida da rampa. Neste momento a robô se desloca novamente para o porto, reiniciando o ciclo, mostrado na Figura 7.

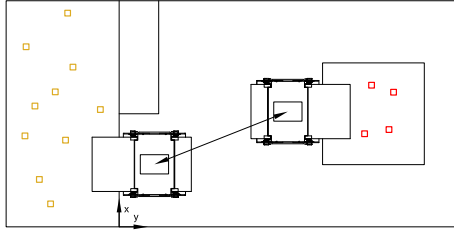


Figura 7: Ciclo de movimentação da Balsa

4.1 Controle Constante e PID

O objetivo do controle da orientação é manter a Balsa sempre alinhada com as paredes laterais do cenário, isto é, manter $e_{\Delta}(t) = 0$ (Equação 1). Isto garante que a leitura dos sensores atendam aos requisitos para que o veículo conheça sua localização no ambiente da competição.

O controle da orientação da Balsa é feito por controladores Proporcional-Integral-Derivativo, sobreposto a um controle constante de deslocamento. A partir da configuração dos atuadores mostrada anteriormente na Figura 6, seja

$$e_{\Delta}(t) = l_2(t) - l_1(t) \quad (1)$$

Tem-se que o sinal de controle, $u_{x,y}$, é constante ou zero, Eq. 2 e 3, atuando de forma a promover o deslocamento acionando as bombas opostas à direção desejada em um valor K .

$$u_{+x,+y}(t) = \begin{cases} K_{+x,+y}, e_{x,y}(t) > 0 \\ 0, e_{x,y}(t) < 0 \end{cases} \quad (2)$$

$$u_{-x,-y}(t) = \begin{cases} 0, e_{x,y}(t) > 0 \\ K_{-x,-y}, e_{x,y}(t) < 0 \end{cases} \quad (3)$$

Utilizando-se sobreposição, o controlador responsável pelo ângulo fornece através de um modelo PID o sinal de controle u_{Δ} conforme a equação 4, onde estão associados os ganhos proporcional, K_p , integral, K_i e derivativo, K_d .

$$u_{\Delta}(t) = K_p e_{\Delta}(t) + K_i \int_0^t e_{\Delta}(t) dt + K_d \frac{de_{\Delta}(t)}{dt} \quad (4)$$

A saída do PWM, sendo arredondada para um inteiro de 0 a 255 utiliza a equação 5 para cada uma das 8 bombas. Como o sinal de controle de rotação deve atuar de forma diferencial, este é associado às bombas, quando ímpares, tem seu valor somado ao valor de referência do controle de posição, já quando pares, tem seu valor subtraído. Isto permite que estas gerem um momento em torno do centro de gravidade do robô, rotacionando-o simultaneamente a realização do posicionamento.

$$PWM_i(t) = \begin{cases} u_{\pm x, \pm y}(t) + u_{\Delta}(t), i = 1, 3, 5, 7 \\ u_{\pm x, \pm y}(t) - u_{\Delta}(t), i = 2, 4, 6, 8 \end{cases} \quad (5)$$

5 Resultados

Os dois veículos realizam suas funções devidamente para a execução da prova, mantendo uma comunicação a fim de garantir a perfeita realização da tarefa. Desta forma são apresentados os gráficos de deslocamento. Na Figura 8 pode-se identificar o funcionamento do algoritmo para busca de barris. Este é o procedimento demonstrado previamente neste texto, onde a Empilhadeira sai da Balsa, localiza o cubo, segue até sua posição, o recolhe e segue para o seu carregamento na Balsa. Todo esse procedimento acontece utilizando a visão computacional em conjunto com o algoritmo de odometria.

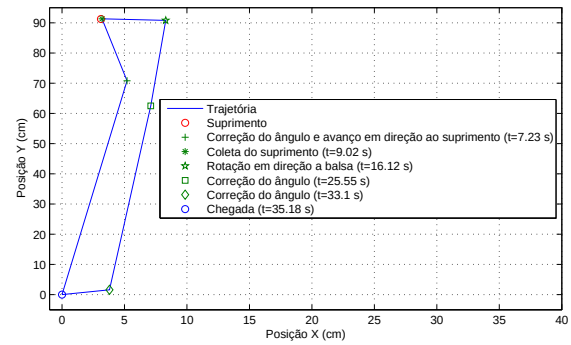


Figura 8: Resultado experimental da movimentação da Empilhadeira durante a busca por barris

Na Figura 9 pode-se observar o sistema de controle de posição em funcionamento. Neste é visto como o deslocamento em y ocorre de forma gradual e suave. Em x tem-se uma movimentação menos linear, principalmente pela ação dos elementos de controle de ângulo e leituras incorretas do sensor na superfície da água, que ocasionam o cálculo incorreto do erro. Mesmo com estes fatores, nota-se que o transportador realiza sua tarefa com eficiência.

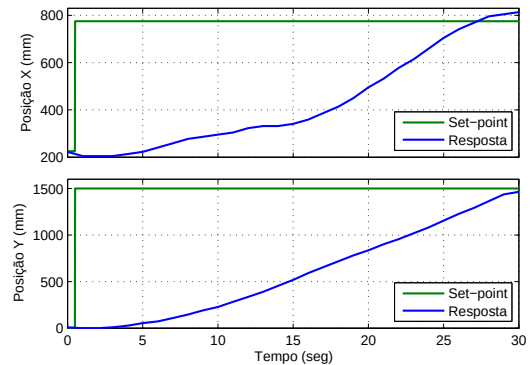


Figura 9: Resultado experimental da movimentação da Balsa durante transporte

O procedimento de subida da Empilhadeira é mostrado na Figura 10, com a vista da faixa de alinhamento e centralização, que garante o correto posicionamento para a realização do transporte.

Na descida da Empilhadeira, após a atracagem da Balsa, segue seu procedimento de saída da Balsa e logo inicia a busca e o retorno transportando um novo bloco.

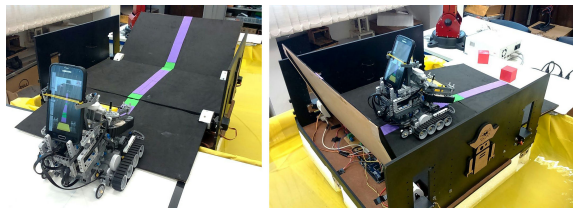


Figura 10: Embarque e desembarque da Empilhadeira na Balsa

A condição de paralelismo à parede do tanque é vista no transporte na Figura 11, onde pode-se ver o deslocamento do veículo com a manutenção do ângulo em relação à borda, o que garante que a posição seja facilmente calculada utilizando os sensores disponíveis.

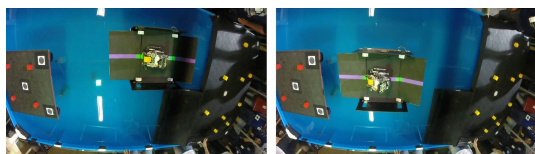


Figura 11: Procedimento de deslocamento da Balsa

Para uma melhor compreensão destes resultados o leitor é convidado a assistir o vídeo que originou estes, através do link: <https://youtu.be/kwe0E1LtaoM>

6 Conclusões

As definições de projeto, baseadas em uma metodologia que visava prever e promover soluções para as variações que normalmente seriam encontrados neste tipo de desafio, resultou em um projeto sólido e robusto, garantindo que o conjunto de robôs fosse capaz de cumprir a prova de maneira eficiente e mais consoante possível com a realidade.

O uso diferenciado de visão computacional em um dispositivo móvel proporcionou um sistema de sensoriamento e detecção de objetos que foi um dos pontos cruciais no bom desenvolvimento obtido. Este recurso proporcionou flexibilidade na programação dos algoritmos de controle do sistema.

O sistema de locomoção da Balsa também foi um ponto forte no projeto, pois garante a capacidade de deslocamento omnidirecional, facilitando manobras. Este modo de propulsão exige um maior gasto computacional, porém garante juntamente com os sensores de ultrassom a correta localização do robô no campo da prova. É importante ressaltar a grande importância tecnológica deste projeto. Foram desenvolvidos diversos

recursos que podem, e serão, aplicados em trabalhos futuros, colaborando com recursos para solucionar novos problemas, seja na área de odometria, controle de angulação, visão computacional, eletrônica e sistemas embarcados terrestres e marítimos.

Agradecimentos

Os autores agradecem ao CEFET-MG / Divinópolis por viabilizar o trabalho e a todos os membros do Grupo de Estudos em Robótica pelo apoio.

Referências

- Alami, R. (1996). A fleet of autonomous and cooperative mobile robots, *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, Vol. 3, pp. 1112–1117 vol.3.
- Asama, H., Sato, M., Goto, N., Kaetsu, H., Matsumoto, A. and Endo, I. (1996). Mutual transportation of cooperative mobile robots using forklift mechanisms, *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, Vol. 2, pp. 1754–1759 vol.2.
- LeJOS, Java for Lego Mindstorms (n.d.). <http://www.lejos.org/>.
- Lew, M. W., Horton, T. B. and Sherriff, M. S. (2010). Using lego mindstorms nxt and lejos in an advanced software engineering course, *Software Engineering Education and Training (CSEE T), 2010 23rd IEEE Conference on*, pp. 121–128.
- Ozaki, K., Asama, H., Ishida, Y., Matsumoto, A., Yokota, K., Kaetsu, H. and Endo, I. (1993). Synchronized motion by multiple mobile robots using communication, *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*, Vol. 2, pp. 1164–1169 vol.2.
- Pessin, G. (2013). *Estratégias inteligentes aplicadas em robôs móveis autônomos e em coordenação de grupos de robôs*, PhD thesis, Universidade de São Paulo, São Carlos. Tese de Doutorado.
- Stilwell, D. J. and Bay, J. S. (1993). Toward the development of a material transport system using swarms of ant-like robots, *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pp. 766–771 vol.1.