

# UMA ABORDAGEM GENERALISTA PARA O PROBLEMA DE PLANEJAMENTO E CONTROLE DE TRAJETÓRIA DE MÚLTIPLOS ROBÔS TERRESTRES EM AMBIENTES DINÂMICOS

JOHNATHAN FERCHER DA ROSA\*, PAULO FERNANDO FERREIRA ROSA\*

*\*IME - Instituto Militar e Engenharia  
Praça Gen. Tibúrcio, Praia Vermelha, Rio de Janeiro, RJ, Brasil*

Email: johnathanfercher22@gmail.com, rpaulo@ime.eb.br

**Abstract**— In this work, is presented an approach to deal with the problem of multiple path planning and control. Is sought to facilitate the integration with different robot types and increase the security in navigation. The approach focus in the resolution of the problem in environments which has mobile obstacles and in the execution in real environments. Was obtained a good rate of resolutions, however, is necessary to improve the obstacle avoidance.

**Keywords**— Path Planning, Multiple Robots, Real-time, Collision Avoidance, Dynamic Environments.

**Resumo**— Nesse trabalho, é apresentado uma abordagem de resolução do problema de planejamento e controle de trajetória de múltiplos robôs. É buscado facilitar a integração de diferentes tipos de robôs e aumentar a segurança na movimentação. A abordagem foca na resolução do problema em ambientes com existência de obstáculos móveis e execução em ambientes reais. Foi obtido uma boa taxa de resoluções, entretanto, é necessário aprimorar o desvio de obstáculos.

**Palavras-chave**— Planejamento de Trajetória, Múltiplos Robôs, Tempo real, Evitar Colisões, Ambientes Dinâmicos.

## 1 Introdução

O tema tratado neste artigo faz parte da base necessária de qualquer plataforma de robótica móvel. Para que um robô consiga se locomover em um ambiente, com finalidade de resolver um determinado problema, é necessário dividir e analisar a questão em subproblemas.

O primeiro subproblema que deve ser resolvido é: encontrar o caminho que um determinado robô deverá seguir. Ou seja, dado um determinado espaço de trabalho limitado e que pode possuir áreas não-convexas, deve-se encontrar uma sequência de pontos que conectem duas localizações. Na figura 1 encontra-se um exemplo do problema, onde as limitações do espaço de trabalho são representadas por áreas pretas nas extremidades, os obstáculos fixos são representados por áreas pretas dentro do espaço de trabalho, as localizações que devem ser conectadas estão representadas por ( $q_{init}$ ) e ( $q_{goal}$ ) e o caminho encontrado é representado por uma sequência de pontos conectados.

Esse problema é conhecido como planejamento de trajetória, e vem sendo objeto de estudo por várias décadas. Existem inúmeros algoritmos para a resolução desse problema, alguns focam em obter a menor trajetória, outros uma trajetória que conserve melhor a dinâmica dos robôs e outros simplesmente buscam obter uma trajetória. Esses diferentes propósitos se devem a existência de diversas situações.

A questão principal do problema de planejamento de trajetória é o nível de conhecimento que se tem sobre o ambiente de atuação. A situação

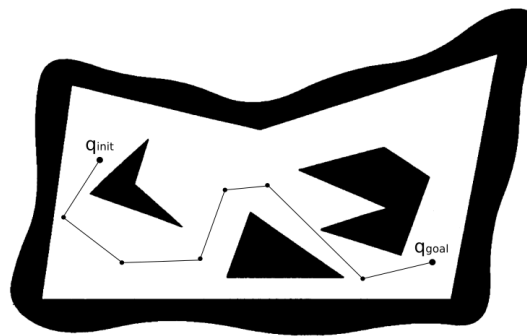


Figura 1: Espaço de trabalho limitado com obstáculos fixos. O caminho encontrado é representado por uma sequência de pontos conectados.

ideal para resolver o problema é aquela que possui as seguintes três características: há conhecimento de todos os obstáculos do ambiente, há conhecimento da localização e do formato de todos os obstáculos e não existem obstáculos móveis;

As primeiras abordagens, como o algoritmo *Roadmap* (Latombe, 2012), consideravam resolver o problema nessas situações ideais, porém, a medida que o problema se aproxima de uma situação real, as abordagens devem ser adaptadas, e muitas vezes encontrar um caminho simples, já se torna uma grande questão. Em outros casos, como na presença de objetos móveis, o planejamento de trajetória deve ser dividido em etapas, pois um caminho encontrado no início de uma execução pode ser invalidado pelos obstáculos móveis.

O segundo subproblema que deve ser tratado, é fazer com que um determinado robô siga um caminho. Quando se busca resolver o problema

em um ambiente real é necessário aplicar diversas modificações na solução, pois, como pode ser visto em (Nourbaskhsh and Siegwart, 2004) diferentes robôs, possuem diferentes limitações de cinemática e dinâmica. Dois exemplos de plataformas se encontram na figura 2.

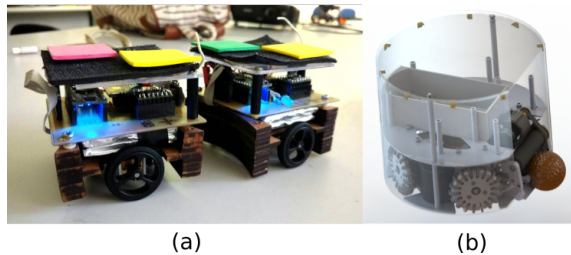


Figura 2: Duas plataformas de futebol de robôs diferentes. Na figura (a), dois robôs diferenciais do SIRLab e na figura (b), um robô omnidirecional do RoboIME.

Plataformas diferenciais, como as presentes na figura 2.a, são conhecidas por possuírem uma baixa manobrabilidade e serem não-holonômicas; pois as únicas duas rodas da plataforma são paralelas, o que impossibilita que o robô se mova em qualquer direção. Plataformas omnidirecionais, como a presente na figura 2.b são conhecidas por possuírem uma alta manobrabilidade e serem holonômicas; suas 4 rodas suecas de 90 graus possibilitam que a mesma se movimente em qualquer direção. Em geral, uma plataforma omnidirecional possui vantagens na movimentação, porém, em comparação com uma plataforma diferencial, a mesma possui um controle mais difícil de ser realizado e está mais suscetível a derrapagens. Existem diversos modelos de movimentação para robôs terrestres sobre rodas, e cada um possui suas particularidades.

### 1.1 Motivação

É notável que a robótica não está mais presa à ambientes controlados como chão de fábrica. Por isso, esse trabalho se concentra em ser funcional em ambiente dinâmicos. Recentemente, cada vez mais aplicações que rodam à céu aberto vêm surgindo, a exemplo disso, temos os carros autônomos. Qualquer aplicação que seja executada em ambientes como esse, deve ser capaz de se adaptar a existência de incertezas no ambiente, seja no sensoramento ou na presença de objetos móveis no ambiente.

Existem diversos problemas que podem ser resolvidos com a utilização de robôs de forma mais eficiente e segura em comparação à seres humanos. Além disso, aplicações que utilizam múltiplos robôs vêm se tornando cada vez mais comuns, pois, em alguns casos a utilização de um conjunto de robôs para resolver um determinado problema, pode ser a única forma de resolução, e em outros

casos, tais resoluções apresentam vantagens como, redundância e divisão de trabalho.

Com finalidade de possibilitar o controle de um conjunto de robôs terrestres em ambientes dinâmicos, foi criado esse *framework* (Rosa and Rosa, 2016) que terá seu funcionamento descrito nesse artigo. O *framework* controla um conjunto de robôs homogêneos e pode ser facilmente adaptado para controlar diferentes tipos de plataformas.

Apesar do problema ser próximo ao enfrentado por equipes de futebol de robôs, o objetivo do artigo é resolvê-lo em um nível mais geral, não existindo preocupação com os aspectos de futebol.

### 1.2 Organização do trabalho

Esse trabalho está organizado em mais 5 seções. Na seção II, são apresentados os trabalhos relacionados. Na seção III, é formulado o problema. Na seção IV, é apresentado a metodologia. Na seção V, são apresentados os experimentos e resultados. Por último, na seção VI, são apresentados a conclusão e trabalhos futuros.

## 2 Trabalhos Relacionados

Existem muitos trabalhos relacionados a planejamento de trajetória, porém, poucos buscam resolver o problema em ambiente real utilizando múltiplos robôs em um ambiente dinâmico. Isso se deve ao fato que, apesar de parecer simples, o problema abrange uma grande quantidade de variáveis, por isso, muitas abordagens são incapazes de serem resolvidas em ambientes dinâmicos, pois possuem um crescimento do custo computacional exponencial e podem se tornar intratáveis, como pode ser visto em (Surynek, 2010).

Em geral, para a resolução do problema em ambientes dinâmicos, é considerado que todos os objetos possuem uma movimentação conhecida e previsível, assim, são planejadas trajetórias para os robôs do conjunto de forma a evitar colisões com obstáculos em todo o tempo de execução. Apesar de funcionar em simulações, essa abordagem impossibilita a utilização em ambientes reais, pois, além de distanciar a solução da realidade, plataformas robóticas estão sujeitas a inúmeros erros de atuação e sensoramento, o que faz com que dificilmente uma plataforma consiga seguir um caminho com extrema precisão, mesmo em um chão aderente.

Em (Motee et al., 2010) é abordado o problema de planejamento de trajetória de múltiplos robôs em ambientes dinâmicos simulados, a questão é tratada como um problema de otimização onde existe preocupação dos caminhos saírem de uma posição inicial e irem para uma posição objetivo, de forma a evitar colisões com obstáculos e suavizar o caminho encontrado.

Para que seja possível o algoritmo encontrar caminhos livres de colisões em qualquer instante, o mesmo considera que os obstáculos possuem uma movimentação conhecida e que os robôs são capazes de seguir os caminhos sem que nenhum tipo de erro ocorra. Com essas restrições são obtidos bons resultados, porém, tal abordagem não funcionaria em ambientes reais.

Em (Chiang et al., 2015) é abordado o problema de planejamento de trajetória de um único robô em ambientes altamente dinâmicos simulados. As simulações ocorrem em ambientes com diferentes formatos de área de trabalho e estão presentes cerca de 300 obstáculos fixos e dinâmicos.

O autor divide a abordagem de resolução do problema em duas etapas, uma de pré-execução e outra e correções retroativas. Durante a primeira etapa, é calculado uma trajetória que busca evitar colisões com obstáculos fixos do ambiente. Após obter uma trajetória, o algoritmo guia o robô ao longo do caminho utilizando a técnica de campos potenciais artificiais (*APF*) (Goodrich, 2002). A abordagem resulta em uma boa taxa de resoluções e evita colisões. Porém, a mesma se concentra apenas em simulações onde o robô controlado não possui restrições de cinemática e dinâmica, o que em ambientes reais deve ser considerado.

Em (Rodrigues, 2013), é abordado o problema de construção de um time de futebol de robôs para a categoria *Small Size League (SSL)*. São abordados vários tópicos relacionados ao problema, inclusive planejamento e controle de trajetória de múltiplos robôs.

Para a resolução do problema, o autor adaptou o algoritmo de planejamento de trajetória *Rapidly-Exploring Random Trees (RRT)* (LaValle, 1998), para considerar os limites de movimentação da plataforma robótica utilizada e a arquitetura de inteligência da equipe de futebol de robôs. O algoritmo nomeado de *Behavioral Kinodynamic Balanced Growth Tree (BK-BGT)*, apesar de resolver bem o problema no ambiente de futebol de robôs, é uma abordagem muito específica e gera um custo computacional mais elevado.

### 3 Formulação do Problema

O ambiente definido é bastante complexo, pois sofre influência de inúmeras variáveis e pode ser definido, como: Seja um espaço euclidiano  $W$ , denominado espaço de trabalho e representado por  $\mathbb{R}^2$ , que pode possuir qualquer configuração espacial, podendo conter áreas não-convexas, seja a posição e orientação que um robô pode ocupar em  $W$  representada por uma pose  $p = [x \ y \ \theta]^T$ , seja uma trajetória contínua que ligue duas poses representada por uma sequência de poses  $P = \{p_1, p_2, \dots, p_n\}$ , um robô representado por  $r = [pose \ raio]^T$ , um obstáculo representado por

$o = [x \ y \ raio]^T$  e um estado do espaço de trabalho representado por  $E = \{R, O\}$  onde  $R = \{r_1, r_2, \dots, r_n\}$  é um conjunto de robôs controláveis e  $O$  o conjunto de obstáculos, assumindo que a forma de  $R$  e  $O$  são conhecidas e que  $O$  possui obstáculos fixos e móveis. O problema pode ser definido como:

**Para cada  $r_n \in R$ , encontrar uma trajetória  $P_n$  que ligue a pose atual  $p_n^a$  e a pose objetivo  $p_n^o$ , de maneira a evitar contato com  $O$  e com os demais robôs em  $R$ .**

Na figura 3, encontra-se um exemplo do ambiente onde o problema é tratado. Na imagem, os limites do espaço de trabalho são representados pelos segmentos de reta aos extremos, obstáculos fixos são representados por quadrados pretos, obstáculos dinâmicos são representados por círculos em cinza e os robôs são representados pelos quadrados nas cores vermelho, verde e azul (com suas respectivas áreas de objetivo e caminhos da mesma cor).

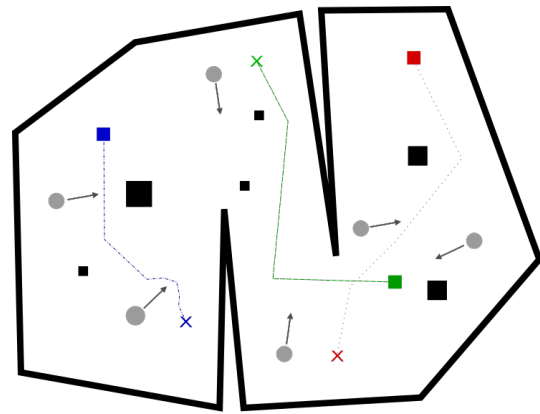


Figura 3: Exemplo de ambiente onde o problema é tratado. A área de atuação é limitada e existem obstáculos fixos e dinâmicos.

## 4 Metodologia

Para os experimentos, foi utilizado uma plataforma de futebol de robôs da categoria *IEEE Very Small Size Soccer (VSS)*. Essa categoria utiliza uma câmera no topo do campo para o rastreamento dos objetos, assim, tem-se uma diminuição na complexidade do problema, pois sempre é conhecido a posição de todos os objetos no campo.

O planejamento e controle de trajetória ocorrem em duas etapas diferentes. Na primeira etapa, denominada etapa (*offline*) é executado o planejamento de trajetória e na última etapa (*online*) é executado o controle de trajetória.

### 4.1 Planejamento de trajetória

Para realizar o planejamento de trajetória, foi utilizado a biblioteca *Open Motion Planning Library*

(OMPL) (Şucan et al., 2012). A biblioteca provê diversos algoritmos de planejamento de trajetória, desde algoritmos clássicos até algoritmos do estado da arte.

Apesar do ambiente experimental ser simples e possuir poucos obstáculos, na primeira etapa, foi optado pela utilização do algoritmo *RRT* para realizar o planejamento de trajetória dos robôs, de forma a levá-los para uma determinada localização e somente evitar colisões com obstáculos fixos. Cabendo o desvio entre robôs e de obstáculos móveis na última etapa de execução. O algoritmo *RRT*, é conhecido por resolver problemas de alta complexidade, como pode ser visto em (LaValle, 1998), por isso, sua utilização é importante, pois não limita a abordagem para aplicações de baixa dimensão.



Figura 4: Exemplo de ambiente onde o problema é tratado. A área de atuação é limitada e existem obstáculos fixos e dinâmicos.

Na figura 4, os robôs com um etiquetas amarela no topo da carenagem são os controláveis pela abordagem. Nesse caso, a cor da etiqueta secundária de cada robô serve como representação do caminho que será traçado pelo mesmo, sendo verde e roxo na imagem.

#### 4.2 Controle de trajetória

O controle de trajetória é executado utilizando a técnica *APF*. No ambiente, é definido que todos os robôs e obstáculos no espaço de trabalho possuem a mesma carga e áreas de interesse, possuem carga oposta. Assim, naturalmente os robôs se deslocam para as áreas de interesse ao mesmo tempo que evitam contato entre si e com obstáculos.

Cada robô é guiado por um caminho ( $P$ ) que parte da pose inicial do robôs ( $p_1$ ) e se direciona para a pose objetivo ( $p_n$ ). Sempre é calculado a força atrativa ( $\vec{F}_a$ ) para o ponto suprajacente à localização do robô. Caso exista um obstáculo próximo ao robô é calculado a força repulsiva ( $\vec{F}_r$ ) em relação ao mesmo e as forças de atração e re-

pulsão são somadas resultando na força resultante ( $\vec{F}$ ), que será aplicada sobre a plataforma.

Nas figuras 5 e 6, encontra-se o funcionamento da técnica. Na figura 5, o robô é representado por um círculo azul, o obstáculo é representado por um círculo cinza, o campo atrativo é representado por um círculo verde em ( $p_2$ ), o campo repulsivo é representado por um círculo vermelho no obstáculo e as forças atuantes no robô é representada pelo vetor ( $\vec{F}$ ), sendo esse a soma vetorial de ( $\vec{F}_a$ ) e ( $\vec{F}_r$ ).

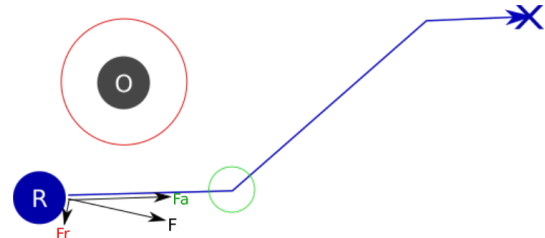


Figura 5: Técnica de campos potenciais artificiais aplicados ao guia de um robô.

A técnica de campos potenciais artificiais é executada retroativamente, isto é, em um ciclo. Quando os robôs chegam a uma determinada distância da pose ( $p_i$ ), os mesmos passam o campo atrativo para a pose ( $p_i + 1$ ). Dessa forma, tem-se o efeito na trajetória apresentado na figura 6, onde a trajetória calculada pelo algoritmo *RRT*, encontra-se em azul escuro e o caminho que um robô executaria utilizando a técnica *APF*, encontra-se em azul claro. Nota-se um pequeno desvio da trajetória entre ( $p_1$ ) e ( $p_2$ ), isso se deve à presença de um obstáculo próximo.

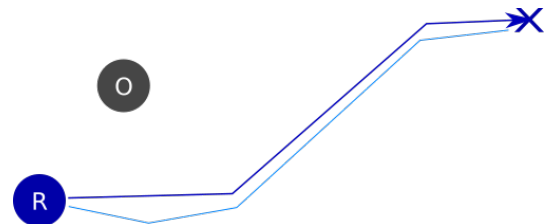


Figura 6: Efeito da técnica campos potenciais artificiais ao guiar um robô.

É necessário a combinação do algoritmo *RRT* com a técnica *APF*, pois ambos são complementares para o problema definido. Somente com o *RRT* não é possível desviar de objetos móveis e somente com o *APF* não é possível resolver o problema em ambientes mais complexos.

Para aplicar a técnica *APF* no controle de robôs reais ou modelados próximos a realidade, deve-se converter a força resultante da técnica *APF* para comandos dos atuadores. Para realizar essa conversão, pode-se utilizar um modelo cinemático da plataforma utilizada. Caso seja utilizado uma plataforma diferencial, pode-



se utilizar o modelo cinemático apresentado em (Blažič, 2014), para realizar a conversão.

Na figura 7, encontra-se a vista superior de uma plataforma diferencial, onde  $w$  é uma velocidade angular,  $v$  é uma velocidade linear,  $B$  é comprimento do eixo e  $\theta$  é orientação do robô.

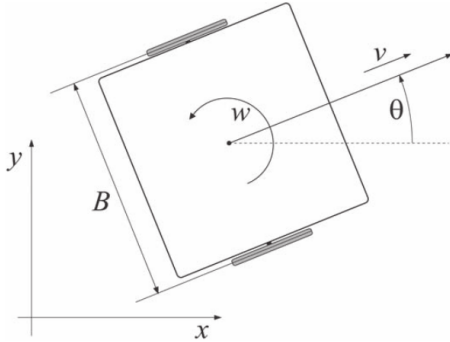


Figura 7: Vista superior de uma plataforma robótica diferencial.

Para controlar a plataforma, existe a necessidade de encontrar  $v$  e  $w$ , que façam com que o robô se direcionem para a força resultante  $\vec{F} = [x \ y \ \theta]$ . Com esse propósito, é utilizado a equação 1, onde  $\dot{p}$  está relacionado com  $\vec{F}$ .

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (1)$$

Os valores  $v$  e  $w$  são utilizados nas equações 2 e 3 para obter as velocidades das rodas direita ( $v_R$ ) e esquerda ( $v_L$ ).

$$v_R = v + \frac{w_B}{2} \quad (2)$$

$$v_L = v - \frac{w_B}{2} \quad (3)$$

Como o modelo utilizado é uma aproximação, existem erros na execução, por isso, os valores de  $v$  e  $w$  são constantemente corrigidos com base no comportamento da plataforma e os valores de ( $v_R$ ) e ( $v_L$ ) são enviados para os robôs utilizando rádio frequência. Como todo sistema está sujeito a erros, é utilizado a técnica de controle *PID* (Ogata, 2003) para reduzir a influência desses erros.

## 5 Experimentos

Foram realizados vários experimentos, tanto em ambiente simulado quanto em ambiente real. Em todos os experimentos foi utilizado o *SDK* (VSS-SDK, 2016), que provê programas úteis para executar simulações e experimentos com robôs reais.

Inicialmente, foram feitos 40 experimentos em ambientes simulados, onde eram controlados 3

robôs diferenciais em um ambiente com mais três obstáculos. 20 experimentos foram realizados em ambientes estáticos, onde os obstáculos eram fixos e 20 experimentos foram realizados em ambientes dinâmicos, onde os obstáculos se movimentavam de forma aleatória.

Cada experimento possuía um tempo de execução de 1 minuto, durante esse tempo eram computados: número de caminhos resolvidos, quantidade de colisões entre robôs do conjunto e quantidade de colisões com obstáculos. Para constatar que a abordagem utilizada de fato reduz a quantidade de colisões, foi implementado uma solução inspirada na movimentação de um time de futebol de robôs, onde não se busca evitar colisões, apenas se focam em chegar a um determinado local, essa solução foi nomeada de comparativo.

Nas tabelas 1, 2 e 3, encontra-se a média dos resultados obtidos nos experimentos. Nas tabelas, **Resol.** representa o número de caminhos resolvidos, **C. Robôs** representa o número de colisões entre robôs e **C. Obstác.** representa o número de colisões com obstáculos.

Tabela 1: Resultados simulados em ambientes estáticos.

	Resol.	C. Robôs	C. Obstác.
<b>Comparativo</b>	50.2	27.7	23.8
<b>Framework</b>	56.2	1.7	0.6

Os dados da tabela 1 são referentes aos experimentos em ambientes estáticos, onde os obstáculos eram fixos. Nota-se a partir da mesma, que apesar da solução comparativa não buscar evitar colisões, assim, não perdendo tempo desviando de obstáculos, a mesma resolve menos caminhos do que o *framework* proposto, devido a quantidade de colisões que ocorrem. O *framework* reduz significativamente o número de colisões em relação a solução comparativa, assim aumentando a segurança dos robôs.

Tabela 2: Resultados simulados em ambientes dinâmicos.

	Resol.	C. Robôs	C. Obstác.
<b>Comparativo</b>	30	39.6	44.6
<b>Framework</b>	57.3	2.8	4.9

Nota-se, a partir da tabela 2, que a diferença entre resoluções de caminhos aumenta significativamente em ambientes dinâmicos, devido ao drástico aumento de colisões sofridas pela solução comparativa, enquanto o *framework* mantém resultados próximos.

Além das simulações, foram feitos 10 experimentos em ambiente real utilizando a plataforma SirSoccer. Os robôs eram rastreados utilizando o sistema de visão computacional do SDK (VSS-SDK, 2016). Na figura 8, encontra-se o ambiente de experimentos, os padrões de cores que possuem etiquetas azuis eram considerados obstáculos.

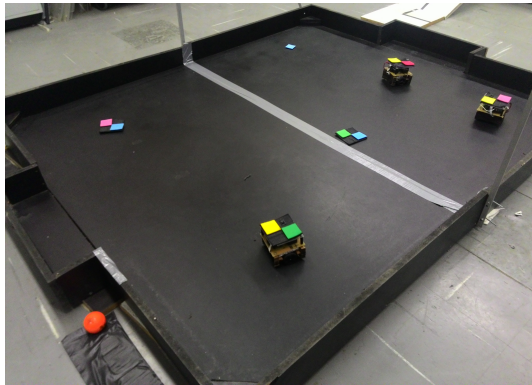


Figura 8: Ambiente de experimentos.

O resultado obtido nos experimentos em ambiente real com obstáculos estáticos, encontra-se na tabela 3. Nota-se um resultado próximo aos obtidos nas simulações, como é possível ver nos resultados das tabelas 1 e 2.

Tabela 3: Resultados obtidos em ambientes reais estáticos.

	Resol.	C. Robôs	C. Obstác.
Framework	55.5	3.5	2.7

## 6 Conclusões e Trabalhos Futuros

A abordagem se demonstra funcional em controlar um conjunto de robôs terrestres. Além de tornar a movimentação dos robôs mais segura, a mesma consegue resolver mais caminhos do que uma abordagem que não se preocupa em evitar colisões.

Contudo, ainda ocorrem colisões. Para trabalhos futuros, será realizado o refinamento do modelo dos robôs, incorporando a dinâmica dos mesmos.

Como a abordagem facilita a inclusão de diferentes robôs com diferentes modelos de movimentação, também é de interesse que se possua um conjunto heterogêneo de robôs trabalhando no mesmo ambiente.

## Agradecimentos

Agradecimentos a Demanda Social/CAPES, a equipe RoboIME do Laboratório de Laboratório de Robótica e Inteligência Computacional do IME, e a equipe SirSoccer do Laboratório de Sistemas Inteligentes e Robótica (SIRLab) da FAETERJ.

## Referências

Blažič, S. (2014). On periodic control laws for mobile robots, *IEEE Transactions on Industrial Electronics* **61**(7): 3660–3670.

Chiang, H.-T., Malone, N., Lesser, K., Oishi, M. and Tapia, L. (2015). Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments, *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, IEEE, pp. 2347–2354.

Goodrich, M. A. (2002). Potential fields tutorial, *Citeseer* **1**. 22 mar. de 2016.

URL: 1

Latombe, J.-C. (2012). *Robot motion planning*, Vol. 124, Springer Science & Business Media.

LaValle, S. M. (1998). Rapidly-exploring random trees a new tool for path planning, *Ames* **1**. 09 nov. de 2016.

URL: 1

Motee, N., Jadbabaie, A. and Pappas, G. (2010). Path planning for multiple robots: An alternative duality approach, *American Control Conference (ACC), 2010*, IEEE, pp. 1611–1616.

Nourbakhsh, I. and Siegwart, R. (2004). *Introduction to autonomous mobile robots*, The MIT Press, Cambridge, Massachusetts, England, ISBN 0.

Ogata, K. (2003). *Modern control engineering*, Prentice Hall PTR.

Rodrigues, S. H. (2013). *Sistemas Autônomos e Inteligentes para Robôs Cooperativos no Ambiente Small Size League*, PhD thesis, Instituto Militar de Engenharia, Rio de Janeiro.

Rosa, J. F. and Rosa, P. F. F. (2016). Construção de um framework de planejamento e controle de trajetória em tempo real de múltiplos robôs terrestres, *III Workshop on MSc Dissertation and PhD Thesis in Robotics, ROBÓTICA*. 2016-12-01.

URL: 1

Şucan, I. A., Moll, M. and Kavraki, L. E. (2012). The Open Motion Planning Library, *IEEE Robotics & Automation Magazine* **19**(4): 72–82.

Surynek, P. (2010). An optimization variant of multi-robot path planning is intractable., *AAAI*, p. 6.

VSS-SDK (2016). Um projeto opensource criado pela comunidade que auxilia novas equipes a ingressarem na competição de futebol de robôs ieev very small size [soccer]. 05 out. de 2016.

URL: 1