

UMA ABORDAGEM HÍBRIDA PARA NAVEGAÇÃO EM AMBIENTES DINÂMICOS BASEADA NA DRRT

STEPHANIE KAMARRY*, PHILLIPE CARDOSO*, LUCAS MOLINA*, ELYSON ÁDAN NUNES CARVALHO*,
EDUARDO OLIVEIRA FREIRE*

**Grupo de Pesquisa em Robótica da UFS (GPRUFS)
Departamento de Engenharia Elétrica da Universidade Federal de Sergipe (DEL/UFES)
Av. Marechal Rondon, São Cristóvão, SE, Brasil*

Emails: skamarry@gprufs.org, phillipe@gprufs.org, lmolina@ufs.br, ecarvalho@ufs.br,
efreire@ufs.br

Abstract— In this paper it is proposed a hybrid approach to mobile robot navigation in dynamic environments based on the DRRT planning algorithm and a reactive control obstacle avoidance with no need of replanning. This is possible because of DRRT characteristics of building a compact representation of the environment, enabling the robot to go from any point to the destination. This planning is switched with a reactive obstacle avoidance control according to the necessity of the system by a supervisor automaton. The proposed methodology was tested in simulation environment with maps that typically generate local minimum problems for reactive navigations combined with moving obstacles that causes problems to deliberative approaches. Results demonstrate the method applicability.

Keywords— RRT, DRRT, Robotics, Hybrid Navigation.

Resumo— Neste trabalho é proposta uma abordagem híbrida para navegação de robôs móveis em ambientes dinâmicos baseada no algoritmo de planejamento da DRRT e um controle reativo de desvio de obstáculos sem que haja a necessidade de um replanejamento. Isso é possível graças as características da DRRT que constrói uma representação compacta do ambiente de modo que possibilita o robô sair de qualquer ponto e chegar ao destino. Esse planejamento é chaveado com um controle reativo de desvio de obstáculos de acordo com a necessidade do sistema através da supervisão de um autômato. A metodologia proposta foi testada em ambiente de simulação com mapas que normalmente geram problemas de mínimo local para navegações reativas aliados a obstáculos móveis que impossibilitam a aplicação de abordagens deliberativas. Os resultados obtidos demonstram a aplicabilidade do método.

Palavras-chave— RRT, DRRT, Robótica, Navegação Híbrida.

1 Introdução

Uma suposição comum em algoritmos de planejamento de movimento é que o ambiente é bem definido, de tal forma que a localização do robô em relação aos obstáculos e a posição de destino são conhecidos (Latombe, 2012). Esta suposição é válida em ambientes estáticos, no entanto, em ambientes reais, o processo de planejamento de movimento para navegação autônoma de um robô móvel envolve a superação de vários desafios como alterações não modeladas no ambiente estático, a presença de obstáculos dinâmicos, ou mesmo a presença de outros robôs no ambiente, cujas ações e objetivos nem sempre são conhecidas pelo sistema de planejamento.

Em geral, a assunção de um ambiente estático bem definido não se sustenta. Existe um grau de incerteza que surge como resultado de erros de detecção e de ruído, além da imprecisão dos atuadores e outros fatores incontroláveis, como a patinação das rodas. Estes desafios são amplificados quando o espaço de planejamento possui dimensão elevada.

Métodos de planejamento baseados em amostragem (ou aproximados), apesar de não representarem fielmente o espaço de interesse, possibilitam uma representação mais compacta do mesmo,

o que torna possível a aplicação de técnicas de planejamento em espaços de dimensão elevada em tempo real (Latombe, 2012). Entre os planejadores de movimento baseados em amostragem, estão os planejadores probabilísticos, que têm como seus maiores representantes o PRM (do inglês, *probabilistic road map*) (Kavraki et al., 1996) e a RRT (do inglês, *Rapidly-exploring Random Tree*) (LaValle and Kuffner, 2001).

A RRT foi apresentada por LaValle como uma estrutura de dados eficiente que visa realizar uma busca em espaços com grandes dimensões. A ideia chave é tender em direção à exploração de partes inexploradas do ambiente através da amostragem aleatória de pontos no espaço de trabalho. Por ser probabilisticamente completa sob condições gerais, o sucesso do desempenho da RRT em muitos problemas de planejamento de movimento tem levado a grandes extensões e aplicações desta abordagem, até mesmo em ambientes com obstáculos móveis.

Planejamento de movimento em ambientes dinâmicos e incertos é um problema importante na área de navegação de robôs móveis. Uma maneira de abordar o problema de planejamento dinâmico é simplesmente re-executar o algoritmo RRT original para cada iteração de planejamento (Zucker et al., 2007). Porém, ainda que exista a chance

de que todos os obstáculos e a posição de destino tenham sido movidos, é bastante possível que as alterações no ambiente sejam pequenas o suficiente para que o plano da iteração anterior ainda seja válido. Neste caso, o custo computacional extra de construção de um novo planejamento a partir do zero é ineficiente (Frazzoli, 2001).

A observação de que a informação de planejamento de iterações anteriores deve ser preservada levou ao desenvolvimento de três modificações no algoritmo da RRT: ERRT (do inglês, *Execution Extended RRT*) (Bruce and Veloso, 2006), RRT Dinâmica (Ferguson et al., 2006) e RRF (do inglês, *Reconfigurable Random Forests*) (Li and Shie, 2002).

A ERRT é frequentemente citada como o primeiro algoritmo desenvolvido para ser utilizado em ambientes dinâmicos em tempo real (Bruce and Veloso, 2006). O algoritmo da ERRT mantém uma única árvore e caso durante a navegação do robô seja detectada uma possibilidade de colisão, a árvore é descartada e outra é construída. Além disso, quando um planejamento é gerado com sucesso, os estados ao longo do caminho da configuração inicial (q_{init}) até a configuração de destino (q_{goal}) são inseridos em um *waypoint cache* que será utilizado para polarizar o sorteio de novas amostras nas próximas interações de planejamento. Este algoritmo é motivado pelo pressuposto de que, se o algoritmo é atualizado com uma frequência elevada, uma pequena porcentagem da árvore original tem de ser modificada porque os estados que foram utilizados em um planejamento previamente bem sucedido tendem a ser bons estados intermediários no plano atual.

RRT Dinâmica baseia-se na ideia de que é mais eficiente para o planejamento reparar a árvore existente, do que, reconstruir inteiramente uma nova árvore. Esta abordagem primeiramente gera uma RRT padrão a partir de uma configuração inicial e um configuração de meta. Quando ocorrem alterações no espaço de configurações, todas as partes da RRT que são invalidadas por estas mudanças são cortadas. Ao contrário da ERRT, apenas as configurações que colidem e seus nós “filhos” são descartados de maneira eficiente. A RRT Dinâmica utiliza a mesma técnica de polarização de sorteio aplicada na ERRT. No entanto, ela supera ERRT por reparar a árvore.

Crescer várias árvores para trás a partir de um conjunto de configurações de destino tem se mostrado eficiente no contexto do planejamento *one-shot* para manipuladores redundantes (Hirano et al., 2005). O algoritmo RRF constrói uma floresta de RRTs desconectadas para efeito de planejamento de consulta múltipla, com o objetivo de incrementalmente construir uma estrutura de dados que, eventualmente, capta toda a topologia do espaço livre.

Além dos métodos já citados, outros algo-

ritos já foram propostos para buscar solucionar o problema (Ferguson et al., 2006) (Aguinaga et al., 2008) (Otte and Frazzoli, 2015). Porém, apesar da ampla variedade de abordagens para planejamento de movimento dinâmico, ainda não há uma solução geralmente aplicável ao problema. Alguns dos inconvenientes das atuais abordagens incluem dificuldade em replanejar em espaços com alta dimensionalidade, planejamento com restrições não-holonômicas ou diferenciais, planejamento com obstáculos móveis e eficiente tratamento das alterações de estado inicial e final.

Neste trabalho é proposta uma abordagem híbrida para o problema de planejamento de movimento em ambientes dinâmicos e/ou incertos. Para isso, a etapa de planejamento é feita baseada em uma técnica variante da RRT, a DRRT (do inglês, *Dispersion RRT*) (Kamarré et al., 2015). Na tarefa de rastreamento da trajetória gerada pela DRRT, o robô pode, se necessário, navegar guiado conjuntamente por um controlador reativo para desvio de obstáculos dinâmicos. O sistema de navegação do robô é supervisionado por um autômato responsável por determinar qual ação deve ser tomada.

Isso só é possível devido à característica de construção da DRRT, em que o objetivo não é encontrar um caminho entre dois pontos, mas obter uma representação compacta do ambiente, com um menor custo computacional. Com isso, uma árvore é gerada de forma que é possível partir de qualquer ponto do ambiente e chegar ao destino. Este é o ponto chave para a possibilidade de aplicação para ambientes dinâmicos sem a necessidade de replanejamento, uma vez que a abordagem deliberativa da DRRT pode ser chaveada com uma abordagem reativa sempre que um obstáculo que não estava no planejamento inicial é encontrado na trajetória do robô, fazendo com que o mesmo desvie do obstáculo e logo em seguida já se conecte a outro nó que o levará à posição de destino.

Este trabalho está organizado da seguinte forma: Na seção 2 é apresentada a técnica de planejamento utilizada. A abordagem proposta é detalhada na seção 3. Os resultados comparativos são discutidos na seção 4. Por fim são apresentadas as conclusões na seção 6, seguida pelas referências bibliográficas.

2 DRRT: Uma nova abordagem para aumentar a dispersão dos nós na RRT

A RRT clássica possui muita redundância de informação no que concerne à quantidade de nós representando um pequeno espaço, ou seja, mais de um nó cobrindo uma determinada área do ambiente. Para resolver esse problema a DRRT propõe o acréscimo de uma nova regra de seleção dos pontos, de forma a aumentar a cobertura do ambiente para uma mesma quantidade de nós em compara-

ção com a RRT clássica.

Para cada nó é definida uma área que é caracterizada como restrição de sorteio de acordo com a função $C(q_i, r)$, esta função gera um círculo de raio r e centro na configuração q_i . Para uma árvore de tamanho de aresta fixo (d) o raio é definido como o próprio tamanho do passo (Figura 1).

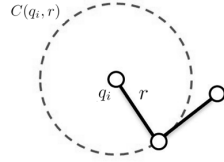


Figura 1: Círculo da restrição de sorteio.

Aplicando essa regra para todos os nós pertencentes à árvore e fazendo a união dos círculos resultantes é encontrada a área de restrição de sorteio (Figura 2). Assim, pontos sorteados dentro da área de restrição são descartados da mesma forma que pontos que geram colisão com obstáculos.

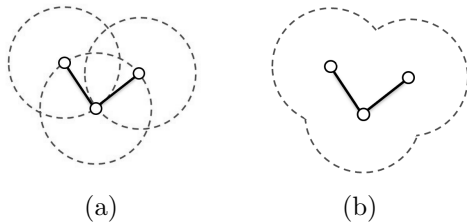


Figura 2: (a) Aplicação da função $C(q_i, r)$ para todos os nós; (b) Área de restrição obtida através da união de (a).

3 Abordagem Proposta

O algoritmo RRT clássico exige informação inicial perfeita relativa ao ambiente; se novas informações são detectadas após a geração do planejamento, um novo plano tem de ser gerado a partir do zero. Na navegação de robôs móveis que envolve ambientes dinâmicos, a necessidade de mudanças no planejamento original é bastante comum. Nesses casos, efetuar o replanejamento do caminho pode ser computacionalmente custoso e, em muitos casos, intratável visto que enquanto a nova árvore é criada o mapa pode ser modificado novamente (Elbanhawi and Simic, 2014).

Diante deste problema é proposta a criação de uma árvore enraizada na posição de destino e que represente, de forma compacta, todo o ambiente de navegação, a DRRT. No trabalho original é apresentada uma medida de dispersão que permite avaliar a ocupação do ambiente pela DRRT. Essa medida foi utilizada para crescer uma árvore a partir do destino até que a mesma ocupe 100% do ambiente. Dessa forma, é possível que o robô

saia de qualquer parte do mapa e chegue à posição desejada, a menos que um obstáculo que não foi considerado surja no meio da trajetória. Nessas situações, é necessário uma postura reativa do robô para evitar a colisão.

Para resolver esse problema, um autômato governado por eventos discretos é modelado para trabalhar como supervisor da navegação do robô de forma a chavear a arquitetura de navegação sempre que um obstáculo, que não estava inicialmente previsto, é detectado.

O autômato desenvolvido pode ser visto na figura 3 e é formado pelos eventos da tabela 1. Nele o sistema é iniciado após o planejamento criado pela DRRT. A partir daí, o robô primeiramente é conectado ao nó mais próximo (evento n) e então segue o galho (evento c_1) até que o destino, que é o galho mais próximo, seja alcançado (evento x). Caso algum obstáculo que não estava no planejamento apareça no caminho do robô (evento o), um controlador reativo de desvio de obstáculos é ativado (evento c_2) até que o robô consiga desviar do mesmo (evento d). No momento em que os obstáculos que não foram previstos deixam de ser detectados pelo robô, é provável que o robô não se encontre mais no galho o qual ele estava seguindo. Em outras abordagens, esse seria o momento em que um novo planejamento seria executado. No entanto, como a DRRT gera uma representação completa do mapa, o robô precisa apenas reconectar-se ao nó mais próximo (evento n) e seguir o novo galho até que o destino final seja alcançado.

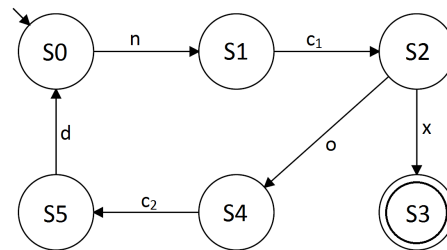


Figura 3: Autômato supervisor da navegação do robô.

Tabela 1: Descrição dos eventos do autômato supervisor

Evento	Descrição
n	Conectar ao nó mais próximo
c_1	Ativar controlador para seguir o galho
x	Robô alcançou o ponto de destino
o	Obstáculo detectado
c_2	Ativar controlador de evitar obstáculos
d	Robô desviou do obstáculo

Para que o robô siga a trajetória gerada pela

árvore foi desenvolvido um controlador de posição final em que o nó conectado ao ponto no qual o robô se encontra é considerado, temporariamente, como posição desejada. Esse processo se repete de forma que o robô percorra todo o galho e chegue ao ponto de destino.

O controlador utilizado foi feito com base no apresentado em (Secchi, 1999) e contém as seguintes equações para as velocidades linear (V_{pf}) e (W_{pf}) angular do robô, respectivamente:

$$V_{pf} = kv_{pf} \tanh(\rho) \cos(\alpha) \quad (1)$$

$$W_{pf} = kw_{pf} \alpha + kv_{pf} \frac{\tanh(\rho)}{\rho} \tanh(\rho) \cos(\alpha) \quad (2)$$

em que kv_{pf} e kw_{pf} são constantes positivas, ρ é a distância entre o sistema de coordenadas do robô e o sistema de coordenadas do destino e α o erro de orientação do robô com relação à linha que liga o robô ao próximo nó que é considerado como ponto de destino temporário.

Com relação ao desvio de obstáculos, foi utilizado um controlador inspirado no desenvolvido em (Freire et al., 2004), que consiste em manter uma velocidade linear constante e controlar a velocidade angular a partir de um ponto de destino virtual mantido a uma distância fixa do robô. Quando nenhum obstáculo é detectado, o ponto de destino virtual é posicionado no eixo x do sistema de coordenadas do robô. Caso contrário, é realizada uma rotação no ponto de destino virtual, de forma que o mesmo seja mantido oposto ao obstáculo, formando um ângulo de 180° entre o novo ponto de destino e o obstáculo, com relação ao robô.

As equações que regem este controle são:

$$V_{eo} = k_1 \quad (3)$$

$$W_{eo} = kw_{eo} \alpha' + kv_{eo} \frac{\tanh(\rho')}{\rho'} \tanh(\rho') \cos(\alpha') \quad (4)$$

em que k_1 é uma constante positiva, α' o erro de orientação do robô com relação ao ponto de destino virtual, ρ' uma distância, mantida constante, entre o robô e o ponto de destino virtual e kw_{eo} e kv_{eo} constantes positivas relativas a este controlador.

Ambos os controladores foram escolhidos por terem estabilidade demonstrada nas suas respectivas referências. Além disso, no que diz respeito ao supervisor, a linguagem gerada pelo autômato é igual a linguagem marcada do mesmo, ou seja, não há nenhum conjunto de eventos que leve o sistema a um estado onde não é mais possível atingir o estado marcado, que é alcançado quando o robô cumpre seu objetivo.

A abordagem proposta foi testada para ambientes dinâmicos e os resultados podem ser vistos na próxima seção.

4 Resultados

Para evidenciar a eficiência do método foram executados testes por simulação através do *software* Matlab R2013a. Nestes experimentos são inseridos elementos dinâmicos (como obstáculos móveis) no ambiente previamente mapeado pela DRRT. A velocidade dos obstáculos foi definida como igual ou inferior à velocidade do robô, sendo essa a única restrição imposta ao movimento dos obstáculos dinâmicos. Além disso, o sistema de navegação do robô não tem nenhum conhecimento prévio do tipo de movimentação executada pelos obstáculos.

A árvore gerada pela DRRT possui tamanho de aresta de 20 cm e é gerada com um afastamento mínimo de 60 cm em relação aos obstáculos. No que diz respeito ao controlador de evitar obstáculos, foi considerado que este deve ser ativado em casos onde um obstáculo é detectado a menos de 60 cm pelos sensores frontais ou a menos de 40 cm pelos sensores laterais. Dessa forma, o robô consegue detectar quando tem algum obstáculo que não estava no planejamento a sua frente ou se aproximando pelas laterais ao mesmo tempo que consegue navegar em corredores sem que as paredes sejam vistas como obstáculos.

Para este experimento foram utilizados dois mapas com dimensão de 800×800 cm. O primeiro mapa é conhecido como ambiente tipo *trap* e é utilizado para evidenciar o maior problema da navegação reativa: mínimo local. No segundo mapa, os obstáculos se movem de modo a bloquear, temporariamente, as únicas passagens possíveis ao robô.

4.1 Experimento 1

Na figura 4 é apresentado o resultado da navegação do robô em um ambiente tipo *trap* utilizando a abordagem proposta, neste ambiente a maior parte das arquiteturas de navegação reativa levaria o robô a ficar preso dentro da configuração “U” de obstáculos. Caso o robô apenas seguisse a trajetória gerada pela DRRT iria colidir com os obstáculos móveis presentes no ambiente. Portanto, as duas arquiteturas de navegação foram combinadas em uma abordagem híbrida onde ocorre um chaveamento entre as mesmas de acordo com a necessidade do sistema. Isso só é possível devido a característica de mapear o ambiente completo da DRRT, sem qualquer conhecimento prévio do ponto de partida do robô, que permite que o robô possa desviar dos obstáculos e partir de qualquer ponto do ambiente para o destino.

No experimento realizado, inicialmente (Figura 4.(a)) o robô navega seguindo a trajetória planejada pela DRRT (em vermelho), ao detectar um obstáculo que não foi previsto na etapa de planejamento (em (b)) o autômato supervisor seleciona o controlador de desviar de obstáculos

permitindo que o robô deixe de seguir a trajetória inicialmente planejada para garantir a segurança (trajetória em azul). O robô se conecta à árvore novamente a partir do momento em que deixa de detectar a presença do obstáculo dinâmico, caso um novo obstáculo seja detectado (em (c)) este processo se repete até o robô chegue ao ponto de destino (em (d)).

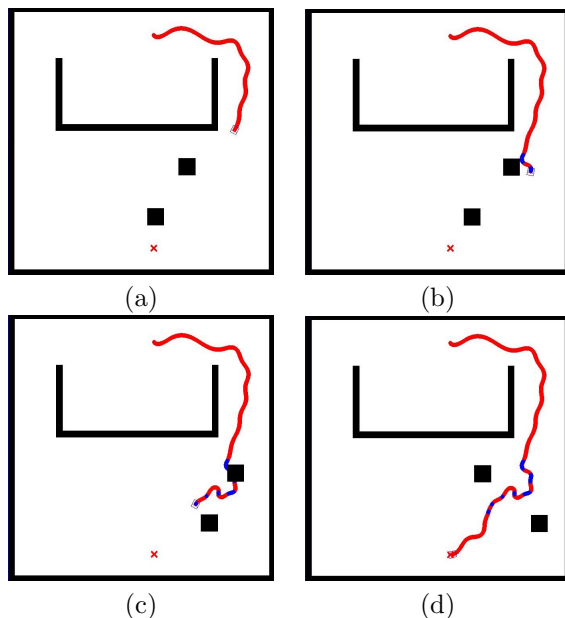


Figura 4: Trajetória executada pelo robô no Experimento 1: (a) O robô navega seguindo a trajetória planejada pela DRRT; (b) e (c) O robô desvia dos obstáculos dinâmicos; (d) o robô chega ao ponto de destino.

A árvore planejada pela DRRT (em verde) e a trajetória executada pelo robô (em vermelho e azul) após a suavização do caminho são apresentadas na Figura 5.

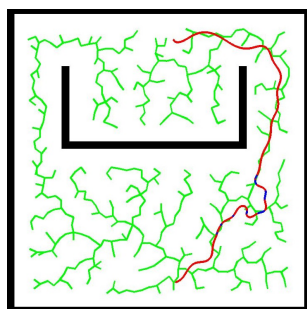


Figura 5: Árvore planejada pela DRRT (em verde) e a trajetória executada pelo robô (em vermelho e azul).

5 Experimento 2

Na figura 6 é apresentado o resultado da navegação do robô em um ambiente onde os obstáculos dinâmicos interrompem temporariamente os

caminhos possíveis que levam o robô ao ponto de destino. Neste tipo de ambiente a estratégia de replanejamento, utilizada por maior parte das abordagens, levaria o robô a execução de uma trajetória muito mais longa, uma vez que ao encontrar um obstáculo no corredor direito do mapa, a nova trajetória seria gerada pelo corredor esquerdo.

Com a aplicação da abordagem proposta, o robô inicia seguindo a árvore em direção ao corredor direito do mapa (Figura 6.(a)), mas ao detectar um obstáculo que não estava previsto no planejamento, o controlador de evitar obstáculo é selecionado pelo supervisor e o robô desvia do obstáculo móvel (Figura 6.(b) e Figura 6.(c)), seguindo até o ponto de destino pelo mesmo corredor onde foi encontrado o obstáculo (Figura 6.(d)).

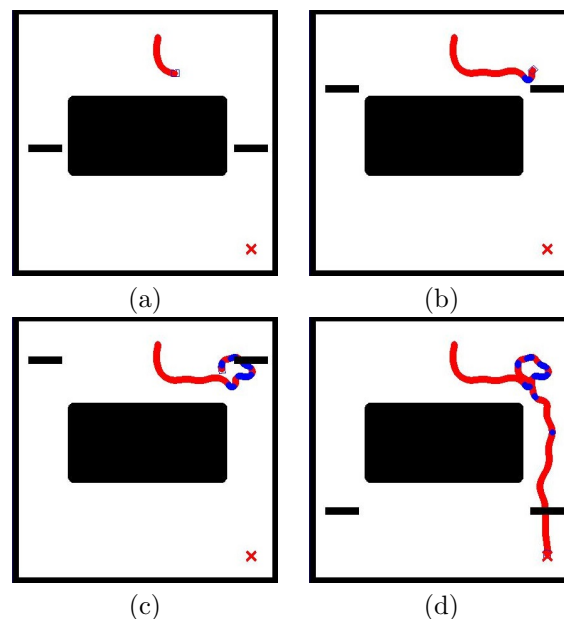


Figura 6: Trajetória executada pelo robô no Experimento 2: (a) O robô navega seguindo a trajetória planejada pela DRRT; (b) e (c) O robô desvia dos obstáculos dinâmicos; (d) o robô chega ao ponto de destino.

A árvore planejada pela DRRT (em verde) e a trajetória executada pelo robô após a suavização do caminho (em vermelho e azul) são apresentadas na Figura 7.

6 Conclusão

Com o objetivo de fornecer soluções de alta qualidade no que concerne ao tempo de execução da tarefa e a segurança do robô em ambientes dinâmicos, foi proposta uma abordagem híbrida através da combinação do algoritmo de planejamento da DRRT com um controlador de evitar obstáculos para criar um algoritmo de navegação em ambientes com obstáculos móveis, sem a necessidade de um replanejamento. Essa abordagem só

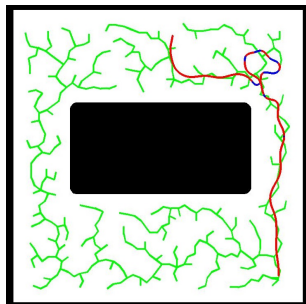


Figura 7:  rvore planejada pela DRRT (em verde) e a trajet ria executada pelo rob  (em vermelho e azul).

foi poss vel devido   caracter stica de representa  o completa e compacta do ambiente da DRRT que permite o rob  partir de qualquer ponto do ambiente e chegar ao ponto de destino.

Como a DRRT mant m apenas uma  rvore, ela oferece uma s rie de vantagens quando comparada com abordagens de replanejamentos no que diz respeito ao problema de planejamento em tempo real. DRRTs s o simples de implementar e envolvem muito poucos par metros, bastando apenas acrescentar a nova regra de restri  o de sorteio de n s   RRT cl ssica e utilizar como crit rio de parada do crescimento da  rvore a densidade de ocupa  o do ambiente.

Os resultados apresentados por simula  o destacam o sucesso do m todo proposto em executar uma tarefa em ambiente din mico, garantindo a seguran a do rob .

Como trabalhos futuros sugere-se a expans o e adapta  o do m todo proposto para ambientes inexatos. Al m disso,   preciso fazer a an lise das situa  es em que os obst culos obstruem o galho principal da DRRT de modo a impossibilitar a conclus o da tarefa.

7 Agradecimentos

Os autores gostariam de agradecer ao CNPq e   FAPITEC pelo apoio financeiro que tornou poss vel a realiza  o desse trabalho.

Refer ncias

Aguinaga, I., Borro, D. and Matey, L. (2008). Parallel rrt-based path planning for selective disassembly planning, *The International Journal of Advanced Manufacturing Technology* **36**(11-12): 1221–1233.

Bruce, J. R. and Veloso, M. M. (2006). Safe multirobot navigation within dynamics constraints, *Proceedings of the IEEE* **94**(7): 1398–1411.

Elbanhawi, M. and Simic, M. (2014). Sampling-based robot motion planning: A review, *IEEE Access* **2**: 56–77.

Ferguson, D., Kalra, N. and Stentz, A. (2006). Replanning with rrts, *ICRA*, IEEE, pp. 1243–1248.

Frazzoli, E. (2001). *Robust Hybrid Control for Autonomous Vehicle Motion Planning*, Department of aeronautics and astronautics, Massachusetts Institute of Technology, Cambridge, MA.

Freire, E., Bastos-Filho, T., Sarcinelli-Filho, M. and Carelli, R. (2004). A new mobile robot control approach via fusion of control signals, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **34**(1): 419–429.

Hirano, Y., Kitahama, K.-i. and Yoshizawa, S. (2005). Image-based object recognition and dexterous hand/arm motion planning using rrts for grasping in cluttered scene, *IROS*, IEEE, pp. 2041–2046.

Kamarr , S., Molina, L., Freire, E. O. et al. (2015). Compact rrt: A new approach for guided sampling applied to environment representation and path planning in mobile robotics, *LARS-SBR*, IEEE, pp. 259–264.

Kavraki, L. E.,  vestka, P., Latombe, J.-C. and Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *Robotics and Automation, IEEE Transactions on* **12**(4): 566–580.

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

LaValle, S. M. and Kuffner, J. J. (2001). Randomized kinodynamic planning, *The International Journal of Robotics Research* **20**(5): 378–400.

Li, T.-Y. and Shie, Y.-C. (2002). An incremental learning approach to motion planning with roadmap management, *ICRA*, Vol. 4, IEEE, pp. 3411–3416.

Otte, M. and Frazzoli, E. (2015). Real-time motion planning/replanning for environments with unpredictable obstacles, *Algorithmic Foundations of Robotics XI*, Springer, pp. 461–478.

Secchi (1999). *Control de Veh culos Autoguiados con Realimentaci n Sensorial*, Disertaci n de Maestr a, Universidad Nacional de San Juan.

Zucker, M., Kuffner, J. and Branicky, M. (2007). Multipartite rrts for rapid replanning in dynamic environments, *IEEE ICRA*, pp. 1603–1609.