

RRTを用いた動的環境における移動ロボットのリアルタイム経路計画

松川 翔太^{a)}・横道 政裕^{b)}・甲斐 崇浩^{c)}

An Extension of RRT for Real-Time Path Re-planning of Mobile Robots in Dynamic Environment

Shota MATSUKAWA, Masahiro YOKOMICHI, Takahiro KAI

Abstract

Recently, the application areas of mobile robots have been extended broadly, for instance, from home to planets. In such cases, it is important to ensure both safety and optimality for the movement of the robot. One approach to this problem is offline trajectory planning and reactive obstacle avoidance. The trajectory (path) planning problem has been studied by many researchers and several algorithms have been proposed. In the case of high-dimensional configuration space, it has been shown that the sampling-based algorithms such as RRT (Rapidly-Exploring Random Tree) are effective. The original algorithm has been extended such that the cost is taken into account (RRT*) or that the re-planning is done effectively (such as eRRT, dRRT, and MPRRT). However, many of existing studies about re-planning consider only the cases that the environment is static, or that the initial state of the robot does not change. Thus, they can not be applied to the environments where the obstacles is moving or the dynamics of the robot has uncertainties. In this article, the authors extend the existing RRT and its variants to be able to manage two cases; the continuous change of the environment and the change of the initial configuration of the robot. The effectiveness of the proposed approach is examined numerical simulations.

Keywords: RRT, RRT*, eRRT, dRRT, CT-space

1. はじめに

近年のロボットの活躍は産業分野だけに留まらず、多方面にわたり、ロボットに求める動作やロボットが使用される環境なども複雑になってきている。また、人間の生活環境で使用されるロボットも実用化が進んできている。そのような状況下で使用されるロボットには、不確定かつ動的な環境での安全な動作が要求される。

そのような動作を実現するための一つの手段として、事前に適切な移動経路を計算するオフラインの経路計画と、計画された経路と各時刻においてセンサによって得られた情報に基づいて行動を決定するオンライン障害物回避を使用する方法がある。しかし、環境が不確定であったり、時間的に変化したりする場合にはオフライン経路計画を再度行う必要がある。

本稿では、高次元空間上の経路計画の手法として有効性が示されている RRT¹⁾ (Rapidly exploring Random Tree) とコ

ストに対する最適性を考慮して拡張した手法である

RRT*²⁾を、環境が動的に変化する場合に拡張し、さらに効率的にリアルタイムで再計画するよう拡張する方法を提案する。

2. RRT と拡張

2.1 RRT と RRT*

移動ロボットの経路計画問題には、ロボットの動特性を考慮した動学的経路問題や、制御入力を考慮した制御型経路計画問題などがあるが、本稿では記述を簡単にするために、配置空間上の幾何学的経路計画問題を扱う。ロボットの配置空間を X とし、自由空間を X_{free} とする。また、初期時刻と終端時刻をそれぞれ t_0, t_s とし、初期状態と終了状態をそれぞれ $x_{init} \in X_{free}, x_{goal} \in X_{free}$ とする。RRT では自由空間上の状態をノードとする探索木を、初期状態を根として、ノードをランダムにサンプリングし、隣接したノードを親として直線を用いたエッジを作成することで成長させる (図 1)。そしてこの処理を終了状態と探索木が接続されるまで繰り返す。このとき RRT は確率的に

a) 情報システム工学専攻大学院生

b) 環境ロボティクス学科担当 准教授

c) 教育研究支援技術センター技術職員

完備である、つまり、初期状態と終了状態を接続する経路が存在する場合には、ノード数を無限に増やした時に確率1でそのような経路が生成されることが保証される。さらに RRT*では、サンプリングされたノードに対して、根からのコストが最短となるように親ノードを選択し、さらに、サンプリングされたノードを親としたほうがコストが少なる場合には、近隣のノードの親ノードを変更するという処理を付加する。そして、終了状態と探索木が接続された場合には全コストを評価し、最適経路が得られるまでノードの追加を繰り返す。こうすることで、漸近的最適性、つまり、ノード数を無限に増やしたときに確率1で最適経路が求まることが保証される。

RRT_algorithm(geometric)

```

1 T.init(x_init);
2 for i = 1 to N do
3   x_rand ← RANDOM_STATE();
4   x_near ← NEAREST_NEIGHBOR(x_rand, T);
5   x_new ← x_rand;
6
7
8   T.add_vertex(x_new);
9   T.add_edge(x_near, x_new);
10
11 return T

```

図 1. RRT アルゴリズム.

2.2 再計画問題への拡張

RRT 及び RRT*では、自由空間は既知であると仮定している。一方で、ロボットを作業させるたびに終了状態が変化する場合や、障害物の移動・消失・発生などによって自由空間が変化する場合には、ゼロから探索木を生成するよりは、既存の探索木をもとにして再計画を行うほうが効率的である。また、ロボットが動作している途中で、事前に計画された経路からロボットの状態が大きく外れた場合には、初期状態・初期時刻を変化させてリアルタイムでの再計画を行う必要がある。自由空間が変化するによって、既存の探索木の中のいくつかのノードやエッジが無効となり、探索木が分断される。また、初期状態が変化した場合には、新しい初期最終状態が既存の探索木中になければ、新しい初期状態を根として既存の探索木と接続しなければならない。そして、最終状態が変化した場合には、既存の探索木を再度成長させていく必要がある。

そのような場合に対する RRT の拡張としては、これま

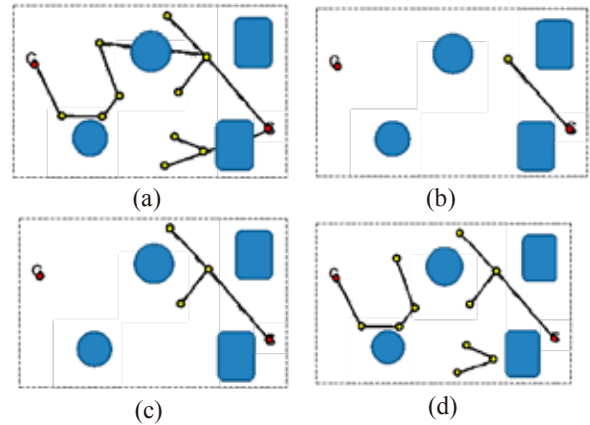


図 2. RRT の再計画.(a) 既存の探索木と追加された障害物.(b) eRRT による初期化.(c) dRRT による初期化.(d) MPRRT による初期化.

で主に自由空間の変化に対する再計画の手法が研究されており、代表的なものとしては eRRT⁴⁾, dRRT⁵⁾, MPRRT⁶⁾ などがある。eRRT では、既存の探索木のうち、解経路のみを残し、その中で根から到達可能な部分のみを残して再計画を行う。また、dRRT では、既存の探索木のうち、根から到達可能な部分のみを残す。一方、MPRRT では、分断された部分木を残して、それらを再接続しながら再計画を行う。これらの手法の概略を図 2 に示す。

3. 動的環境への適用とリアルタイム再計画

本章では、RRT を環境が動的に変化する場合に拡張し、さらにリアルタイム再計画を行う方法について述べる。

3.1 CT-space 上の RRT

障害物が移動することなどによって環境が動的に変化するということは、自由空間 X_{free} が時間的に変化することを意味する。この場合、ある状態 $x \in X$ が X_{free} に属するかどうかは、時間に依存することになる。通常の RRT では、各ノードに明示的に時刻の情報が付随していないため、時刻に関する情報がなければ衝突検出が適切に行えないことになる。このような問題を表現する方法として、CT-space がある。これは、Configuration-Time space の略であり、状態空間と、時間を表す 1 次元実数空間の部分集合 ($I \subset \mathbb{R}$ とする) との直積空間を新たな状態空間と考えるも

のである。この場合、状態 $(x, t) \in X \times I$ には時刻の情報が付随しているため、各時刻における障害物の位置や姿勢などの情報が既知であれば、適切に衝突検出が行えることになる。

通常の RRT においては、サンプリングされた状態から親ノードを選択する際に、その段階での全ノードの中から最も近いものを親ノードとして選択する。これは、経路上の点の時間的な前後関係(因果関係)と状態空間上の方向に依存関係がないからである。しかしながら、CT-space 上では、経路上の点の時間的な前後関係と状態空間内の時間軸上の前後関係は一致しなければならない。このことを説明するために、CT-space 上の時間軸への射影演算子を定義する。 $q = (x, t) \in X \times I$ に対して、 $\tau: X \times I \rightarrow I$ を、 $\tau(q) = t$ として定義する。このとき、CT-space 上のツリーのノードに対しては、ノード q_A がノード q_B の親ノードであった場合、任意のリンク $v = (q_A, q_B) \in V$ に対して

$$\tau(q_A) < \tau(q_B) \quad (1)$$

が成立しなければならない。この制約は、RRT においては親ノードの選択の時点で発生し、RRT* では親ノードの選択と、子ノードとの再接続の際に発生する。

3.2 拡張された RRT と RRT* の再計画

初期状態と初期時刻が変化した場合に再計画を行う RRT および RRT* を、状態空間を CT-space へ拡張する方法について述べる。

状態空間 X を CT-space に拡張した状態空間 $X \times I$ を再度 X と表記する。障害物のない空間を X_{free} とすると、障害物領域 $X_{obs} := X \setminus X_{free}$ となる。簡単のために目標状態 x_{goal} は固定であるとし、 $i \geq 0$ 回目の計画時における初期状態と生成されたツリーをそれぞれ $x_{init}(i)$ 、 $T(i) = \{V(i), E(i)\}$ とする。このとき、 $\forall i_1 < i_2, \tau(x_{init}(i_1)) < \tau(x_{init}(i_2))$ が成立するとする。このことと、式(1)から、再計画を行う際には、前回の計画時に作成されたツリーのうち、根に近いものが無効になることがわかる。このことを考慮して、本研究では eRRT 及び dRRT を基にした再計画法を提案する。

3.2.1 RRT の再計画

i 回目の計画で得られた経路を

$$\begin{aligned} p(i) &= \{x_1^i = x_{init}(i), x_2^i, \dots, x_{m_i}^i = x_{goal}\}; \\ &\forall j \in \{1, \dots, m_i\}, x_j \in X_{free} \\ &\text{and } \forall k < j, \tau(x_k^i) < \tau(x_j^i) \end{aligned}$$

とする。このとき、 $p(i)$ (eRRT の場合) $T(i)$ (dRRT の場合) を用いてステップ $i+1$ における再計画を行う手順について述べる。

直結可能性のチェック

初期状態が変わることによって、目標状態と新しい初期状態を直接接続できる場合がある。そこで処理時間の短縮のために、最初に目標状態と新しい初期状態とを直接接続できるかどうかを確認する。接続できる場合、経路計画は終了となり、

$$\begin{aligned} T(i+1) &= \{V(i+1), E(i+1)\}, \\ V(i+1) &= \{x_{init}(i+1), x_{goal}\}, \\ E(i+1) &= \{(x_{init}(i+1), x_{goal})\} \end{aligned}$$

として次ステップに移る。接続できない場合には、ツリーの再構築を行う。

ツリーの再構築

eRRT に基づく再構築の場合：次ステップに用いるのは生成された経路のみであるので、 $p(i)$ を基にして再構築を行う。 $\tau(x_k^i) > \tau(x_{init}(i+1))$ となる最小の k を k^* として、 $x_{init}(i+1)$ と $x_{k^*}^i$ の接続可能性を確認し、可能であれば経路計画は終了となり、

$$\begin{aligned} V(i+1) &= \{x_{init}(i+1), x_{k^*}^i, x_{k^*+1}^i, \dots, x_{m_i}^i\} \\ E(i+1) &= \\ &\{(x_{init}(i+1), x_{k^*}^i), (x_{k^*}^i, x_{k^*+1}^i), \dots, (x_{m_i-1}^i, x_{m_i}^i)\} \end{aligned}$$

とする。

dRRT に基づく再構築の場合：次ステップに用いるのは、ツリー全体となるので、 $T(i)$ を基にして再構築を行う。 $V(i)$ 内のノードのうち、 $\tau(q) \leq \tau(x_{init}(i))$ となるものを $V_-(i)$ とし、 $E(i)$ のうち、子ノードが $V_-(i)$ に属するものを $E_-(i)$ とすると、これらは過去のものとなるので、 $V(i+1)$ 及び $E(i+1)$ からは除外されるものとなる。ここで、 $V_+ := V(i) \setminus V_-(i)$ 、 $E_+ := E(i) \setminus E_-(i)$ とし、 $V_+(i)$ の中で、 $E_+(i)$ 内に自分を子ノードとするリンクが含まれないものの集合を $V_+^0(i)$ とすると、 $\{V_+(i), E_+(i)\}$ は、 $V_+^0(i)$ の要素を頂点とする部分木の集合となる。そこで、 $V_+^0(i)$ の各要素に対して、 $x_{init}(i+1)$ との接続可能性をチェックし、可能なものを $x_{init}(i+1)$ と接続することでツリーの初期化を行う。初期化されたツリーのノード内に、 x_{goal} が含まれていれば経路計画は終了。なければ次ステップに移る。

サンプリングとツリーの拡張

ここでは前節で説明した手順に従って、サンプリングとノード、リンクの追加を行い、ノード集合内に x_{goal} が含まれていれば経路計画は終了となる。

3.2.2 RRT*の再計画

RRT*での再計画は、基本的にはRRTの場合と同一である。しかしながら、

- 最適解が見つかるまでは何個でも解経路を求める。
- 再構築によって、各ノードのコストも再計算される。点が異なる。最適解の発見については、コスト関数の最小値が既知の場合は、その値を用いて最低性の判定を行うことができるが、一般的にその値は未知であるため、事前に定められた数だけのサンプリングを行ったり、一定時間内でサンプリングを繰り返し行ったりするという方法がとられることになる。

一方で、再構築によるコストの再計算は、以下のように行われる。最初にあるツリー $T=\{V,E\}$ において、根 z_{init} からあるノード $z \in V$ までのコストを $c(z; z_{init})$ とし、また、リンク $e = (z_1, z_2)$ のコストを $\bar{c}(e)$ としたときに、

$$c(z_2; z_{init}) = c(z_1; z_{init}) + \bar{c}(z_1, z_2)$$

の関係があるものとする。Zの中のある経路

$p = \{z_1, z_2, z_3, \dots\}$ に対して根が z_{init} であったときの根からのコストが c_1, c_2, \dots であったときに、根が \bar{z}_{init} に代わることで、それぞれの根から経路の先頭ノードまでのコストが c_1 から c'_1 に変化したとする。

このとき、

$$\begin{aligned} c(z_2; \bar{z}_{init}) &= c(z_1; z_{init}) + (c(z_1; \bar{z}_{init}) - c(z_1; z_{init})) \\ &\quad + \bar{c}(z_1, z_2) = (c'_1 - c_1) + c_2 \\ c(z_3; \bar{z}_{init}) &= (c'_1 - c_1) + c_3 \\ &\quad \vdots \end{aligned}$$

の関係があるので、この関係を用いて新しい根との再接続を行う経路や部分木に対してコストの再計算を再帰的に行えばよい。

4. 実験

前節で提案した手法の有効性を確認するために、数値実験を行う。障害物が移動する環境を考え、RRTについては再計画においてeRRTに基づく方法(A)と、dRRTに基づく

方法(B)の二つの方法を用いる。また、RRT*については、dRRTに基づく方法(C)を用いる。

4.1 問題設定

平面上を移動するロボットを対象とするために、作業空間を R^2 、配置空間を $SE(2) = R^2 \times S^1$ とし、幾何学的問題を解いた。つまり状態空間は配置空間と等しい。自由空間については、

- 作業空間：矩形領域 $(x, y) \in [-6; 6] \times [-6; 6]$

- 動的障害物：半径1で、中心の軌跡は

$$(x_0(t), y_0(t)) = (0, \sin(0.5t))$$

より導出する。また、初期状態及び目標状態を、

$$(x, y, \theta)(0) = (-5, -5, 0), (x, y, \theta)(10) = (5, 5, 0)$$

とした。

この場合、初期状態と目標状態を直結する経路は障害物と衝突するため、回避する経路を生成しなければならない。

また、再計画の時刻と状態を

$$t_1 = 0.1, (x(t_1), y(t_1), \theta(t_1)) = (-4, -4, 0)$$

$$t_2 = 0.4, (x(t_2), y(t_2), \theta(t_2)) = (-3.6, -3, 0)$$

$$t_3 = 0.6, (x(t_3), y(t_3), \theta(t_3)) = (-3, -2, 0)$$

$$t_4 = 6.8, (x(t_4), y(t_4), \theta(t_4)) = (-2.1, -1.7, 0)$$

とした。また、RRT*におけるコスト関数としては、CT-spaceである $SE(2) \times [0; 10]$ 上の L_2 ノルムを用いた各リンクの距離の総和を用いた。

4.2 実験と評価の方法

本稿で提案したアルゴリズムの有効性を評価する項目は下記の二点である。

1. 経路の最適性：生成された経路のコストがどれだけ小さいか。
2. 再計画の効率：経路生成時のノード数と再構築時に再利用されたノード数の比が1に近い。

2.を評価するために、再計画を行う際の状態は、障害物を回避しなければならないような位置にしている。これは、本実験で用いた問題設定においては、再計画時の状態が $x > 1$ となってしまうと、再計画時の初期状態と目標状態を直結することが可能であり、サンプリングすることなく最適解が求まってしまふからである。

各実験において、与えられた初期状態に対して経路計画(再計画)を行う時間の上限は固定(10(sec))とした。方法(A)と(B)では、上限以前に解経路が求められるとその時点で

サンプリングが終了し、方法(C) では、コストの下限値を未知として、上限までサンプリングを続けるようにした。そのため、解が求まるまでの時間は、評価項目には含めていない。

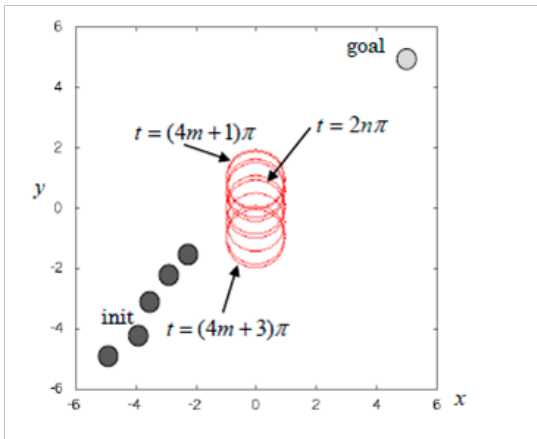


図 3. 作業空間.

手法(A)の結果

手法(A)を用いた経路計画・再計画により生成された経路の例を図 4 に示す。それぞれの再計画において、解経路が求められていることがわかる。図を見ると障害物とぶつかって見えるが、該当する時刻には障害物は矢印で指示している部分にあるので実際には経路と障害物の衝突は起こっていない。

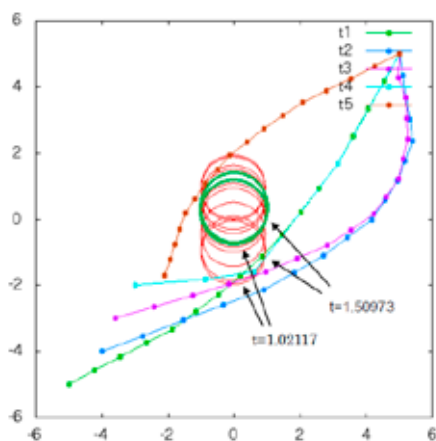


図 4. 手法(A)による結果の例.

手法(B)の結果

手法(B)を用いた経路計画・再計画により生成された経路の例を図 5 に示す。それぞれの再計画において、解経路

が求められていることがわかる。図中の 2 つの矢印の組が示している時刻は、 t_2 と t_5 の経路が障害物と経路がぶつかっているように見える部分の時刻である。図を見ると障害物とぶつかって見えるが、その時刻には障害物は矢印で指示するところにあるので実際には経路と障害物の衝突は起こっていない。

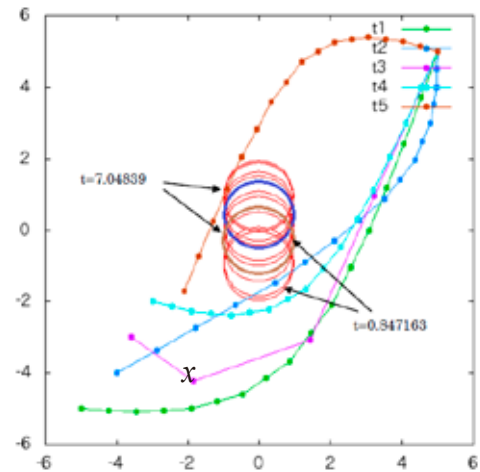


図 5. 手法(B)により生成された経路の例.

手法(C)の結果

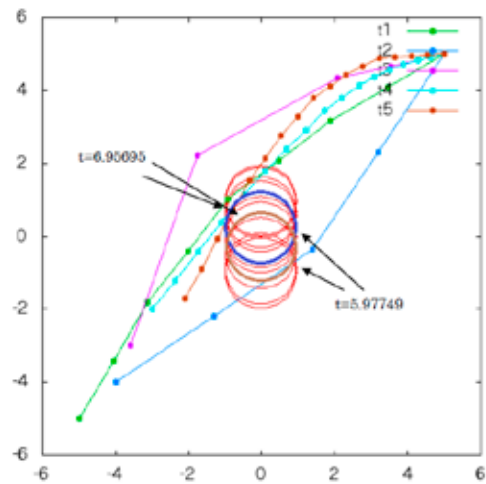


図 6. 手法(C)によって得られた経路の例.

図 6 に手法(C)で得られた経路の例を示す。どの場合においても障害物と衝突しない経路が得られていることがわかる。また、他の方法と比較して、障害物に接近した、総移動距離の少ない経路となっている。

5. 考察

それぞれの手法につき 10 回ずつ経路計画を行った際に、再利用されたノード数、生成されたノード数、そして解経路におけるノード数の平均値を表 1 に示す。

表 1. 経路生成に使用したノードの情報.

	RRTCT(eRRTの場合)	RRTCT(dRRTの場合)	RRTstarCTの場合
初期状態ツリーのノード数			
計画1回目	1	1	1
2回目	6	97	3464
3回目	6	128	4695
4回目	6	173	5439
5回目	1	60	1733
生成されたノード数			
計画1回目	45	72	3624
2回目	22	146	4840
3回目	65	175	5706
4回目	33	217	6289
5回目	27	83	3710
経路のノード数			
計画1回目	15	13	9
2回目	15	15	5
3回目	17	12	7
4回目	17	15	14
5回目	17	20	9

表より、手法(A)と(B)では、再計画に利用されるノード数は絶対値、比率ともに(B)の方が多く、最適化のために経路生成を何度でも繰り返す手法(C) が最も多くのノードを生成し、また再利用するという結果になった。本実験で用いた環境は比較的単純で、経路中のノード数とコストが比例するため、手法(C)の経路におけるノード数が最も少ないという結果が得られた。

6. 結論

本研究では、経路計画の一手法である RRT を、動的障害物が存在するような時変的問題に拡張する方法を提案した。さらに RRT および RRT* を基にしたリアルタイム再計画をおこなうアルゴリズムを提案し、その有効性を数値シミュレーションによって検証した。実験より、経路計画に

多くのノードと計画時間を利用している手法(C)が最も経路に使用しているノード数が少なく、最適解に近い経路を求めていると考えられる。また、手法(A)の方が(B)よりも少ないサンプリング数で経路を求めることができているが、これは障害物が少なく、前回の解経路を再利用することが毎回できたためである。環境がより複雑になった場合は、この関係は逆転するものと考えられる。また、より多くのサンプリングを行うことで、ノード数の少ない経路を求めることができ、最適解に近い解が得られるという傾向が見られた。

今後の課題としては、より現実に近い問題での実験を行うことや、即時的な障害物回避アルゴリズムとの統合などが挙げられる。

参考文献

- 1) S. M. LaValle, Rapidly-Exploring Random Trees: A New Tool for Path Planning, TR 98-11, Computer Science Dept., Iowa State University, October 1998, 1998.
- 2) S. Karaman and E. Frazzoli, Optimal Kinodynamic Motion Planning using Incremental Sampling-based Methods, IEEE Conference on Decision and Control (CDC), Atlanta, GA, 2010.
- 3) S. Karaman, M. Walter, A. Perez, E. Frazzoli, and S. Teller, Real-time Motion Planning using the RRT*, IEEE Conference on Robotics and Automation (ICRA), 2011.
- 4) J. Bruce and M.M. Veloso, Real-Time Randomized Path Planning for Robot Navigation, volume 2752 of Lecture Notes in Computer Science, chapter RoboCup 2002: Robot Soccer World Cup VI, pp. 288-295. Springer, 2003.
- 5) D. Ferguson, N. Kalra, and A. T. Stentz, Replanning with RRTs, Proc. of IEEE ICRA, May, 2006, 2006.
- 6) M. Zucker, J. Kuffner, and M. Branicky, Multipartite RRTs for Rapid Replanning in Dynamic Environments, Proc. of IEEE ICRA, April, 2007, 2007.