



HAL
open science

BoxRRT* -A Reliable Motion Planner

Adina M Panchea, Alexandre Chapoutot, David Filliat

► **To cite this version:**

Adina M Panchea, Alexandre Chapoutot, David Filliat. BoxRRT* -A Reliable Motion Planner. 10th Summer Workshop on Interval Methods, and 3rd International Symposium on Set Membership - Applications, Reliability and Theory, Jun 2017, Manchester, United Kingdom. hal-01574921

HAL Id: hal-01574921

<https://hal.science/hal-01574921>

Submitted on 16 Aug 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

BoxRRT* - A Reliable Motion Planner

Adina M. Panchea¹, Alexandre Chapoutot² and David Filliat²

¹ LIX, Ecole Polytechnique, CNRS, Univ. Paris-Saclay, 91128, Palaiseau, France
panchea@lix.polytechnique.fr

² U2IS, ENSTA ParisTech, Univ. Paris-Saclay, 91120, Palaiseau, France
{chapoutot, filliat}@ensta.fr

Keywords: Planning and control, Robotics, DynIBEX.

Introduction

This work aims at providing a new reliable (asymptotic) optimal motion planner, denoted BoxRRT* which can guarantee a safe path to an unknown initial mobile robot localisation.

Our proposed motion planner is build upon: **i**) methods which use stochastic sampling to discretise the configuration space, *e.g.*, Rapidly-exploring Random Tree, which can guarantee (asymptotic) optimality of the solution, *e.g.*, optimal RRT (RRT*, see for example [2]) and **ii**) modern and new tools [1] for the guaranteed numerical integration.

Problem formulation

Consider the differential system which describes the evolution of a mobile robot system:

$$\dot{\mathbf{s}}(t) = \mathbf{f}(\mathbf{s}(t), \mathbf{u}(t)) \quad (1)$$

with $\mathbf{s} \in \mathbb{S} \subset \mathbb{R}^n$ the measurable state of the system and $\mathbf{u}(t) \in \mathbb{U}$ the admissible control input.

The purpose of the *robust motion planner* is to provide a sequence of control inputs $\mathbf{u} \in \mathbb{U}_{[\mathbf{u}]}^{\Delta t}$ bounded over intervals of time of the form $[K\Delta t, (K+1)\Delta t[$, with $\Delta t > 0$ and $K \in \mathbb{N}$, which will drive the system to reach \mathbb{S}_{goal} from initial states $\mathbf{s} \in \mathbb{S}_{\text{init}}$ while avoiding the

non-admissible states \mathbb{S}_{obs} .

Starting from the formulation given in [3] of such a robust motion planner, there exists a sequence of control input $\mathbf{u} \in \mathbb{U}_{[\mathbf{u}]}^{\Delta t}$ to drive the system from an uncertain initial state to a set of goal states \mathbb{S}_{goal} is as follows

$$\begin{aligned} &\exists K > 0 \text{ and } \mathbf{u} \in \mathbb{U} \text{ such that} \\ &\forall \mathbf{s}_0 \in \mathbb{S}_{\text{init}}, \forall \mathbf{s}(K\Delta t; \mathbf{s}_0) \in \mathbb{S}_{\text{goal}} \text{ and} \\ &\forall t \in [0, K\Delta t], \mathbf{s}(t; \mathbf{s}_0) \in \mathbb{S}_{\text{free}}, \end{aligned} \quad (2)$$

with $\mathbf{s}(t; \mathbf{s}_0)$ the exact solution of (1) from the initial condition \mathbf{s}_0 .

Main results

Let G be the exploration tree, $[\mathbf{s}_{\text{init}}] = \text{Hull}(\mathbb{S}_{\text{init}})$, $[\mathbf{s}_{\text{obs}}] = \text{Hull}(\mathbb{S}_{\text{obs}})$ and $[\mathbf{s}_{\text{goal}}] = \text{Int}(\mathbb{S}_{\text{goal}})$, with $\text{Hull}(\mathbb{S}_{\text{init}})$ the smallest box which contains \mathbb{S}_{init} (*e.g.*, interval hull) and $\text{Int}(\mathbb{S})$ a box included in \mathbb{S} (*e.g.*, inner approximation). The minimal cost from $[\mathbf{s}_1]$ to $[\mathbf{s}_2]$ according to the Hausdorff distance of two intervals (d), is denoted by $\text{cost}([\mathbf{s}_1], [\mathbf{s}_2])$. Let $\text{cost}([\mathbf{s}_1])$ be the total cost to arrive at \mathbf{s}_1 , that is $\text{cost}([\mathbf{s}_1]) = \text{cost}([\mathbf{s}_{\text{init}}], [\mathbf{s}_1])$.

BoxRRT motion planner brief description:*

1. First, G is initialized with the given initial configuration $[\mathbf{s}_{\text{init}}]$.
2. Then, a state $[\mathbf{s}_{\text{rand}}] \in \mathbb{S}_{\text{free}}$ is generated randomly.
3. The tree G is searched for the nearest vertex to $[\mathbf{s}_{\text{rand}}]$ according to a user-defined metric d and the $[\mathbf{s}_{\text{nearest}}]$ vertex is provided.
4. A control input \mathbf{u} is selected according to a desired behaviour. Then, (1) is integrated over a fixed time interval Δt with the initial condition $[\mathbf{s}_{\text{nearest}}]$, to find a new state $[\mathbf{s}_{\text{new}}]$. If the new state and the path between it and $[\mathbf{s}_{\text{nearest}}]$ lie in \mathbb{S}_{free} (*e.g.*, is a collision free path), then $[\mathbf{s}_{\text{new}}]$ is added.

5. Next, the planner tries to find a better parent and children for $[\mathbf{s}_{\text{new}}]$, which needs to provide collision-free-path and a lower cost to and from $[\mathbf{s}_{\text{new}}]$, respectively. For the better parent one searches a set of k -nearest other potential parents to arrive at $[\mathbf{s}_{\text{new}}]$, while for the better children one searches a set of k -nearest other potential children from $[\mathbf{s}_{\text{new}}]$ to other vertices.
6. If a better parent and/or children are found with collision free path and lower cost, than the $[\mathbf{s}_{\text{new}}]$ parent and children information are updated.

These steps are repeated until the algorithm reaches K iterations. Thus, the BoxRRT* algorithm can improve the optimality of the solution, in terms of distance, over time even after the first solution is found.

Application:

The BoxRRT* is performed on the simple car model which involves nonholonomic constraints. The resulted for $K = 20000$ are reported and can be seen in Fig. 1.

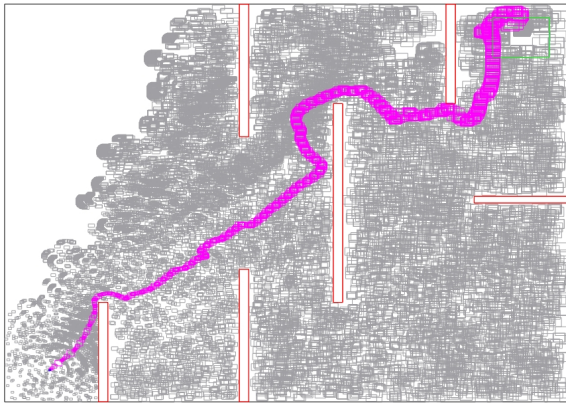


Figure 1: A BoxRRT* solution with $[\mathbf{s}_{\text{init}}]$ the blue box and $[\mathbf{s}_{\text{goal}}]$ the green one while the obstacles are in red: CPU = 745 [s], the number of vertices and the planned path length are 96 and 105 [cm], respectively, while the total number of vertices is 13655.

Acknowledgement

This research was financial supported by the DGA MRIS.

References

- [1] J. ALEXANDRE DIT SANDRETTO AND A. CHAPOUTOT. DynIBEX: a differential constraint library for studying dynamical systems (poster), HSCC, ACM, 2016.
- [2] S. KARAMAN AND E. FRAZZOLI. Optimal kinodynamic motion planning using incremental sampling-based methods. CDC 2010, pages 7681-7687.
- [3] R. PEPY, M. KIEFFER, AND E. WALTER. Reliable robust path planning with application to mobile robots. Int. J. Appl. Math. Comput. Sci., 19(3):413-424, 2009.