IMA Conference on Mathematics of Robotics
9 – 11 September 2015, St Anne's College, University of Oxford



# Path planning on $SE(3)$ for AUVs using the RRT method

**By Jonathan Jamieson & James Biggs**
University of Strathclyde, UK

**Abstract**

This paper presents a kinematic path planning method for a slender autonomous underwater vehicle (AUV) using a technique based on the Rapid-Exploring Random Tree (RRT) algorithm extended to the Euclidean group of motions $SE(3)$. The paths are generated by connecting sub-Riemannian curves on $SE(3)$ that are optimal with respect to a weighted quadratic cost function of the translational and angular velocities subject to the kinematic constraints of the vehicle and are $C^1$ smooth for translational motions. Unlike the majority of other AUV path planning methods, our framework provides a full description of the rotational and translational position as a function of time and the curves satisfy the differential constraints of the vehicle. The modified RRT method is demonstrated in two scenarios, the first with a sparse obstacle field and the second with a dense obstacle field. In both cases the algorithm successfully generates a six degree-of-freedom path from the initial position to a desired final position.

## 1. Introduction

Autonomous Underwater Vehicles (AUVs) are capable of operating for extended periods in conditions and situations that would be hazardous to manned submarines or divers. Unlike Remotely Operated Vehicles (ROVs) that are normally tethered to a larger vessel and piloted by an operator, AUVs are able to navigate and operate without human assistance. Structural inspection, environmental monitoring and surveying are typical applications that underwater robots perform. These tasks require obstacle avoidance algorithms to be implemented on-board to prevent the vehicle from colliding with its environment. Assuming the obstacles are known a-priori, a path can be planned that describes the vehicle's position over time from an initial to final position. Previous research in path planning for AUVs typically focuses on translational position only without planning explicitly the corresponding rotation Alvarez *et al.* 2004; Fu-guang *et al.* 2005. In Neto *et al.* 2010 an RRT based method using Bézier curves was used for two-dimensional path planning. For this paper sub-Riemannian curves are presented that describe translational position and rotation in three dimensions. A tracking controller can then be used to follow the six degree-of-freedom trajectory rather than just the translational path to minimise the tracking error.

The slender underwater vehicle's kinematic motions are planned by assuming it to be a nonholonomic system on $SE(3)$ where for a slender vehicle in motion the lateral motions are quickly damped out. Subject to this constraint, an integral function of translational and angular velocities is minimised to yield a class of paths analogous to sub-Riemannian curves on $SE(3)$. Similar curves have been used for simple wheeled robot path planning on $SE(2)$ (Maclean & Biggs 2013). The shape of the curve is controlled by parameters

of standard functions and optimising these parameters the boundary conditions can be matched. A numerical function minimiser is used to find the parameters that describe a curve from the initial position to a desired final position. Polynomials, b-splines and Bézier curves are all examples of functions used in the literature for trajectory planning in three-dimensional space. Although sub-Riemannian curves offer the advantage of satisfying the nonholonomic constraints, are optimal with respect to a quadratic cost function of the velocities and define a complete six-degree-of freedom motion, one disadvantage of generating trajectories with sub-Riemannian curves is their restricted obstacle avoidance ability due to limited curve reshaping.

To overcome this drawback, this paper uses the sub-Riemannian curves within a well known sampling-based motion planning method, Rapidly-exploring Random Trees (RRT). Lavalle(1998) introduced the RRT method and it has shown to be effective at solving a variety of path planning problems. A "tree" is formed by adding new branches that search through the problem space. The branches represent short trajectories from the existing tree to new positions. Traditionally the trajectories are found using a time-integrated simulation. This can be computationally expensive, especially when motion planning for vehicles with higher degrees of freedom. In our modified RRT algorithm new trajectories are found using sub-Riemannian curves. The format of this paper is as follows. Section 2.1 presents the nonholonomic kinematics of a slender AUV and the reduced form used for motion planning sub-Riemannian curves are given in Section 2.2. The RRT method is introduced in Section 3 and a method for path planning using the curves described in Section 2.2 that avoids obstacles is presented. Section 4 demonstrates the RRT method using sub-Riemannian curves under two scenarios with varying numbers of objects.

## 2. AUV kinematics and sub-Riemannian curves

### 2.1. *AUV Kinematics*

This paper considers long, slender AUVs such as the C-SCOUT (Perrault *et al.* 2003) with the same axis alignment. The vehicle has a translational position in the inertial frame $x = [x_1, x_2, x_3]^T$. The angular velocities about the body axes are $\Omega = [\Omega_1, \Omega_2, \Omega_3]^T$ and the body frame velocities are $v = [v_1, v_2, v_3]^T$. The kinematics can be expressed on the Euclidean group of motions $g \in SE(3)$:

$$g = \begin{pmatrix} 1 & 0 & 0 & 0 \\ x & & R & \end{pmatrix} \tag{2.1}$$

with $R \in SO(3)$ where:

$$\mathrm{SO(3)} \triangleq \{R \in \mathbb{R}^{3\times 3} : \quad R^T R = I \quad \text{and} \quad \det(R) = 1\} \tag{2.2}$$

Since the AUV can only produce thrust in the direction of $v_1$, the lateral motions $v_2$ and $v_3$ are quickly damped out and we wish to minimise the angular rotational velocity $\Omega_1$ therefore the body velocities $v_2 = v_3 = \Omega_1 = 0$ were specified. With these assumptions the kinematics can be expressed a a left-invariant system on the Lie group $SE(3)$:

$$\frac{dg}{dt} = g(v_1 B_1 + A_2\Omega_2 + A_3\Omega_3) \tag{2.3}$$

where $B_1$, $A_2$, $A_3$ are basis elements of the Lie algebra of the Lie group $SE(3)$ (Biggs *et al.* 2007; Biggs & Holderbaum 2009; Walsh *et al.* 1994; Jurdjevic 1996).

### 2.2. *Sub-Riemannian curves for AUVs*

Using the kinematic constraints (2.3) a cost function can be defined that minimises a single weight integral function of angular velocities $\Omega_1$, $\Omega_2$ and the translational velocity $v_1$:

$$J = \frac{1}{2}\int_0^T v_1^2 + c(\Omega_2^2 + \Omega_3^2)dt \tag{2.4}$$

given boundary conditions $g(0) = g_0$ and $g(T) = g_T$ subject to the constraint (2.3), where $c$ is a constant weight and $v_i, \Omega_i$ are measurable and bounded functions. In this case the weight can be manipulated to increase or decrease the angular velocity of the planned reference relative to the translational velocity in the forward direction.

LEMMA 1. *The projection of a particular optimal curve $g_p \in SE(3)$ onto $x_p \in \mathbb{R}^3$ where $x_p = [x_{p1}, x_{p2}, x_{p3}]^T$ and is of the analytic form:*

$$\begin{array}{l} x_{p1} = \frac{c_2^2 v_1}{\gamma}\sin\gamma t + c_1^2 v_1 t \\ x_{p2} = \frac{c_2 v_1}{\gamma}((1-\cos\gamma t)\cos\beta + c_1\sin\beta\,(\gamma t - \sin\gamma t)) \\ x_{p3} = \frac{c_2 v_1}{\gamma}((\cos\gamma t - 1)\sin\beta + c_1\cos\beta\,(\gamma t - \sin\gamma t)) \end{array} \tag{2.5}$$

*where:*

$$\begin{array}{l} \beta = atan2(-M_2(0), M_3(0)) \\ c_1 = \frac{v_1}{\sqrt{r^2+v_1^2}} \\ c_2 = \frac{r}{\sqrt{r^2+v_1^2}} \\ \gamma = \frac{s\sqrt{r^2+v_1^2}}{rc} \\ \alpha = \frac{-sv_1}{cr} \\ s = -\sqrt{M_2(0)^2 + M_3(0)^2} \\ p_3(0) = \frac{p_2(0)M_3(0)}{M_2(0)} \\ r = -\sqrt{p_2(0)^2 + p_3(0)^2} \end{array} \tag{2.6}$$

*and $p_1$, $p_2$, $p_3$, $M_2$ and $M_3$ are the extremal curves which are elements of the dual of the Lie algebra of the Lie group $SE(3)$ (Biggs* et al. *2007; Biggs & Holderbaum 2009; Walsh* et al. *1994; Jurdjevic 1996). Similarly, the projection of $g_p \in SE(3)$ onto $R_p \in S0(3)$ is given by:*

$$R_p = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \tag{2.7}$$

*where:*

$$\begin{array}{l} R_{11} = c_1^2 + c_2^2\cos t\gamma \\ R_{12} = -c_1c_2(\cos t\gamma - 1)\sin(t\alpha+\beta) - c_2\cos(t\alpha+\beta)\sin t\gamma \\ R_{13} = -c_1c_2(\cos t\gamma - 1)\cos(t\alpha+\beta) + c_2\sin(t\alpha+\beta)\sin t\gamma \\ R_{21} = -c1j2(-1+\cos t\gamma)\sin\beta + c_2\cos\beta\sin t\gamma \\ R_{22} = \cos\beta\cos(t\alpha+\beta)\cos t\gamma + (c_2^2 + c_1^2\cos t\gamma)\sin\beta\sin(t\alpha+\beta) - c_1\sin t\alpha\sin t\gamma \\ R_{23} = \cos(t\alpha+\beta)(c_2^2 + c_1^2\cos t\gamma)\sin\beta - \cos\beta\cos t\gamma\sin(t\alpha+\beta) - c_1\cos t\alpha\sin t\gamma \\ R_{31} = -c1j2(-1+\cos t\gamma)\cos\beta - c_2\sin\beta\sin t\gamma \\ R_{32} = -\cos(t\alpha+\beta)\cos(t\gamma\sin\beta + \cos\beta(c_2^2 + c_1^2\cos t\gamma)\sin(t\alpha+\beta) + c_1\cos t\alpha\sin t\gamma \\ R_{33} = \cos\beta\cos(t\alpha+\beta)(c_2^2 + c_1^2 cost\gamma) + \cos t\gamma\sin\beta\sin(t\alpha+\beta) - c_1\sin t\alpha\sin t\gamma \end{array} \tag{2.8}$$

*The initial conditions of the extremal curves will be optimised later to match the boundary conditions on the final position.*

Proof. The method for generating analytical functions for analogous kinematic constraints is described by Biggs & Holderbaum 2010. Also a solution similar to the problem here is given in Jamieson & Biggs 2015 □

The curves described by (2.5) and (2.8) start at the origin, that is $x_p(0) = [0, 0, 0]$ and $R_p(0) = I_d$. The shape of the curve is defined by the values of the parameters in the optimisation vector $\Xi = [M_2(0), M_3(0), p_2(0), T]$ and the user specified values of $v_1$ and $c$. Using a numerical minimiser (such as Matlab's $fmincon$) the optimisation vector can be computed by minimising the cost function in the final position:

$$C_{x_f} = ||x_d(T) - x_p(T)||^2 \tag{2.9}$$

where $x_d = [x_{d1}, x_{d2}, x_{d3}]^T$ is the desired final position relative to the origin in the inertial frame and $T$ is the final time. The only constraint applied to the minimisation is $T > 0$. When the AUV is not at the origin, a suitable $x_d$ must be chosen such that after calculating the optimisation vector the curve can be repositioned so that it reaches the target point from the current position and orientation (Jamieson & Biggs 2015a).

## 3. RRT Path Planning

### 3.1. *The RRT Algorithm*

The standard RRT algorithm is constructed as follows. A starting position is defined with the vehicle's initial position $x(0)$ and orientation $R(0)$ in the inertial frame. The goal position $x_{goal}$ is also chosen. There are two methods for generating target positions ($x_{target}$) to extend the tree, either randomly or a deterministic algorithm that samples positions in the environment. A search is then performed on the existing nodes to determine which is closest to the target node $x_{closest}$ according to some metric. This node is chosen as the point the tree will branch out from during a particular iteration. A segment between this node and the target node is computed using a time integrated simulation of the vehicle. If the segment is deemed acceptable (that is, it doesn't collide with obstacles and is feasible according to any other constraint) then it is added to the tree. A given number of points along that segment are defined as nodes that can be used in future iterations. This basic algorithm works well for vehicles with holonomic constraints. However, for the majority of systems in the real world like AUVs with kinematic constraints the time integration simulation requires a considerable computational time. Since this integration is performed many times the overall computational time is significant, especially for embedded on-board hardware.

### 3.2. *Modified RRT Algorithm*

In this paper the standard RRT algorithm is used, with one significant change. Instead of performing a time integration simulation to generate segments, the curves described in Section 2.2 are used. This is achieved by generating a curve from the closest node to the target node. In doing so, a segment can be generated faster than a time integration simulation. Since the curves are defined analytically, they can be efficiently discretised to check for proximity to obstacles and rejected if a collision is detected.

A biased-random search of the environment was used to generate target nodes. This means that a certain percentage $x_{bias}$ of the target nodes will be the desired final location. The optimum value for $x_{bias}$ is normally obtained through experimentation and is problem specific. If the value is too large, then the environment will not be properly

| | | | | | Final position of the segment | | |
|---|---|---|---|---|---|---|---|
| *Segment* | $M_2(0)$ | $M_3(0)$ | $P_2(0)$ | $T$ | $x_1(T)$ | $x_2(T)$ | $x_3(T)$ |
| 1 | -52.1289 | 34.5024 | 16.3207 | 2.9417 | -1.6309 | 1.5107 | 1.2158 |
| 2 | 97.9715 | 27.6152 | 22.8554 | 2.1514 | -2.1416 | 2.5720 | 2.5373 |
| 3 | 6.9046 | 40.6352 | 2.6240 | 8.8203 | 5.0000 | 5.0000 | 5.0000 |

Table 1: Final trajectory segment values for Scenario 1

explored and tend to get stuck in dead-ends. If $x_{bias}$ is small then it can take a longer time than necessary for the algorithm to converge on the goal state. A simple Euclidean distance metric is used to determine which of the nodes on the tree is closest to the target:

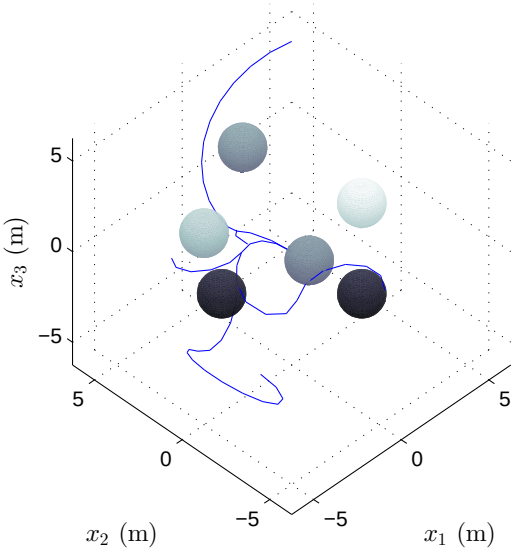
$$d_{euc} = ||x_n - x_{target}||^2 \tag{3.1}$$

where $x_n$ is the position vector of the node on the tree currently being considered. Although this metric is simple to compute, the orientation of the vehicle is not considered. Given the nonholonomic constraints, (3.1) can recommend a node $x_{closest}$ that doesn't result in the shortest path to $x_{target}$. Other metrics have been proposed (Neto *et al.* 2010; LaValle(2006)) but in this paper the basic Euclidean metric is used to demonstrate the method and future work will investigate more sophisticated metrics including those which include the error in rotation.

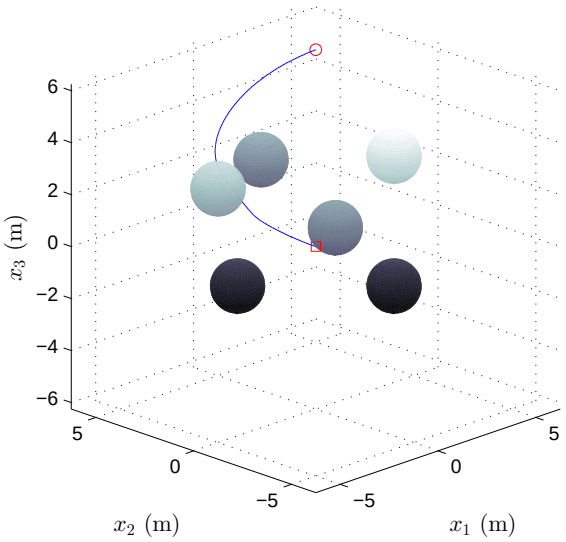
## 4. Simulations

The modified RRT was tested under two scenarios. Scenario 1 has a small density of obstacles and Scenario 2 has a large obstacle field. The obstacles were represented using spheres with any space inside the obstacle volume being considered a region of collision. In both simulations the initial conditions were $x(0) = [0, 0, 0]^T$ and $R(0) = I_d$, the goal position was set as $x_{goal} = [5, 5, 5]^T$ and $x_{bias} = 20\%$. All simulations were performed on a standard $3.4GHz$ desktop computer running Matlab R2012b. The solution time for Scenario 1 was $4.2s$ and a feasible path consisting of 3 segments was found on iteration 42. The entire tree is shown in Figure 2a and the final path is shown in Figure 2b and has a length of $13.9m$. The parameters and final position for each segment are given in Table 1. Scenario 2 had a solution time of $13.4s$ and a total path length $16.7m$. The final path consisting of 4 segments was found on iteration 119 and is shown in Figure 2b. The entire tree at iteration 119 is presented in Figure 2a with all the generated trajectory segments shown. Table 2 gives the parameters and final position for each segment.

Feasible trajectories were found for both scenarios, the solution time was $9s$ faster for the sparse obstacle density field and the final path length was also $2.8m$ shorter. Due to the random nature of the RRT target selection the solutions can vary significantly even with the same initial conditions and object field. For this reason, a more thorough analysis should be performed by running many more simulations and compiling the results together. However, the purpose of this paper is to demonstrate the method.




(a) Complete tree

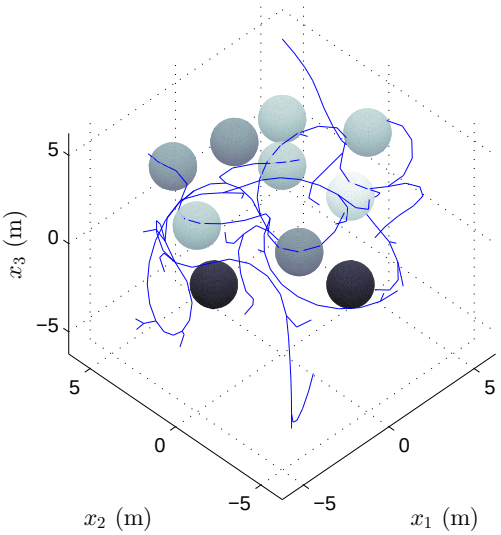



(b) Final trajectory

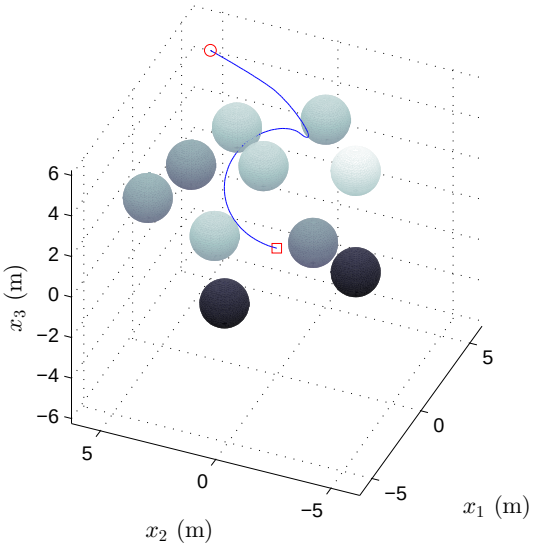
Figure 1: Scenario 1

## 5. Conclusion and future work

This paper presents a modified RRT method that uses sub-Riemannian curves on $SE(3)$ to generate six-degree of freedom paths for AUVs. This class of sub-Riemannian curve is defined using standard functions that satisfy an approximation of a nonholonomic con-




(a) Complete tree




(b) Final trajectory

Figure 2: Scenario 2

straint inherent in the motion of slender vehicles. Two scenarios are used to demonstrate the method in an environment with obstacles. The algorithm is capable of planning a feasible path from an initial to final position in a reasonable computational time. The method is easy to implement, with the main requirement the availability of an efficient numerical function minimiser is the main requirement. The translational velocity along

| | | | | | Final position of the segment | | |
|---|---|---|---|---|---|---|---|
| *Segment* | $M_2(0)$ | $M_3(0)$ | $P_2(0)$ | $T$ | $x_1(T)$ | $x_2(T)$ | $x_3(T)$ |
| 1 | 29.9849 | 25.8332 | 5.6917 | 6.2638 | 4.0917 | 3.1958 | 1.4361 |
| 2 | 50.6622 | 32.3832 | -14.6546 | 2.812 | 4.7560 | 0.9991 | 2.1975 |
| 3 | -38.9249 | 54.1284 | 20.3239 | 4.2816 | 3.9833 | 1.8973 | 4.6866 |
| 4 | 18.0630 | 22.6397 | 3.0029 | 3.3436 | 5.0000 | 5.0000 | 5.0000 |

Table 2: Final trajectory segment values for Scenario 2

the path is $C^1$ smooth but the angular velocities are discontinuous between trajectory segments.

Future work will investigate other metrics for ranking the closeness of two nodes to account for attitude as well as position. Also, a detailed investigation for the best simulation parameters such as the random target bias is needed. This should be performed by running Monte Carlo simulations and testing many randomly generated scenarios with more realistic obstacles. Investigation is also needed to determine if the function that is optimised has multiple local minima or a global minimum to ensure the most appropriate numerical optimiser is used.

## REFERENCES

Alvarez, A., Caiti, A. & Onken, R. (2004) Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *Oceanic Engineering, IEEE Journal of*, **29**, 418–429.

Biggs, J., Holderbaum, W. & Jurdjevic, V. (2007) Singularities of optimal control problems on some 6-d lie groups. *Automatic Control, IEEE Transactions on*, **52**, 1027–1038.

Biggs, J. & Holderbaum, W. 2009 Optimal kinematic control of an autonomous underwater vehicle. *Automatic Control, IEEE Transactions on*, **54**, 1623–1626.

Biggs, J. & Holderbaum, W. (2010) Integrable quadratic hamiltonians on the euclidean group of motions. *Journal of dynamical and control systems*, **16**, 301–317.

Fu-guang, D., Peng, J., Xin-qian, B. & Hong-Jian, W. (2005) Auv local path planning based on virtual potential field. *Mechatronics and Automation, 2005 IEEE International Conference*, vol. 4. IEEE, pp. 1711–1716.

Jamieson, J. & Biggs, J. (2015) Trajectory generation using sub-riemannian curves for quadrotor uavs. *Control Conference (ECC), 2015 European*.

Jamieson, J. & Biggs, J. (2015a) Path planning using concatenated analytically-defined trajectories for quadrotor uavs. *Aerospace MDPI*, **2**, 155.

Jurdjevic, V. (1996) *Geometric Control Theory*. Cambridge University Press. Cambridge Books Online.

Lavalle, S. M. (1998) Rapidly-exploring random trees: A new tool for path planning. *Technical Report*. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.1853`

LaValle, S. M. (2006) *Planning Algorithms*. New York, NY, USA: Cambridge University Press.

Maclean, C. & Biggs, J. D. (2013) Path planning for simple wheeled robots: sub-riemannian and elastic curves on se (2). *Robotica*, **31**, 1285–1297.

Neto, A. A., Macharet, D. G. & Campos, M. F. M. (2010) Feasible rrt-based path planning using seventh order bézier curves. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, pp. 1445–1450.

Perrault, D., Bose, N., OYoung, S. & Williams, C. D. (2003) Sensitivity of auv added mass coefficients to variations in hull and control plane geometry. *Ocean engineering*, **30**, 645–671.

Walsh, G., Montgomery, R. & Sastry, S. (1994) Optimal path planning on matrix lie groups. *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, vol. 2.