2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Down The CLiFF: Flow-aware Trajectory Planning under Motion Pattern Uncertainty

Chittaranjan Srinivas Swaminathan[1], Tomasz Piotr Kucner[1], Martin Magnusson[1], Luigi Palmieri[2] and Achim J. Lilienthal[1]

***Abstract*— In this paper we address the problem of flow-aware trajectory planning in dynamic environments considering flow model uncertainty. Flow-aware planning aims to plan trajectories that adhere to existing flow motion patterns in the environment, with the goal to make robots more efficient, less intrusive and safer. We use a statistical model called CLiFF-map that can map flow patterns for both continuous media and discrete objects. We propose novel cost and biasing functions for an RRT* planning algorithm, which exploits all the information available in the CLiFF-map model, including uncertainties due to flow variability or partial observability. Qualitatively, a benefit of our approach is that it can also be tuned to yield trajectories with different qualities such as exploratory or cautious, depending on application requirements. Quantitatively, we demonstrate that our approach produces more flow-compliant trajectories, compared to two baselines.**

I. Introduction and Motivation

In many practical applications, robots operate in environments that are dynamic due to movement of people, vehicles or other robots. Often the dynamic changes exhibits patterns, which allows us to represent them using statistical models. A robot can benefit from the use of learned information about these patterns in order to generate trajectories that tend to adhere to the flow. For example, a mobile service robot may move with the flow of people or avoid congested areas (to avoid bumping into people) [1], while an unmanned aerial vehicle may move against the flow of air currents (to spend less energy).

In this paper, we address the problem of motion planning in dynamic environments especially considering flow model uncertainty due to partial observability and statistical variance. In particular, we use the Circular Linear Flow Field (CLiFF-map) [2] to represent statistical information about flow patterns. CLiFF-map can be used to model the flow of discrete objects (e.g. people, vehicles or other robots) as well as continuous media (e.g. air). Moreover, CLiFF-map also contains information about how much a region in the environment has been observed and how much motion there is. The CLiFF-map allows us to generate plans based on the learned flow patterns for the entire map and not only for regions where the robot can currently observe motion.

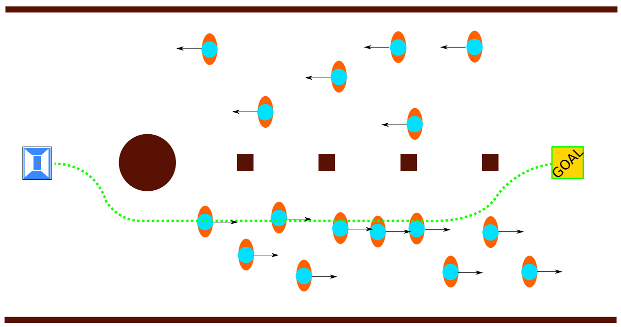


Fig. 1: In the scenario shown, a robot (**blue**) has to reach the other end of a long and crowded airport corridor while considering static obstacles (**brown**), and information regarding the flow of people (**cyan and orange**). The robot can either take the lane below or above. Faced with all this information, we would like the robot to choose the lane below so that it follows the flow of people.

We propose a cost function, called Down-The-CLiFF (DTC) cost, for trajectory planning, which explicitly accounts for the environment's dynamics and the uncertainty in the flow model. Our work is directly inspired by the CLiFF-RRT* algorithm [3]. We discuss how the cost function can be tuned to yield qualitatively different trajectories. For instance, the robot can be tuned to adopt an *exploratory* approach (preferring regions with less information about the dynamics) or a *cautious* approach (preferring regions with more information) to trajectory planning. In this paper we use the DTC cost with the Rapidly exploring Random Tree* (RRT*) algorithm but the cost function is agnostic to the trajectory planning algorithm used.

We implemented and evaluated one instance of the DTC cost function that penalizes motions not adhering to the underlying flows represented in the CLiFF-map (Sec. IV-A). We also propose a biased sampling algorithm (Sec. IV-C). In our implementation we use the DTC cost along with the RRT* algorithm and compare its results with those obtained using two baseline algorithms (Sec. V). The results indicate qualitative and quantitative advantages of our algorithm. Importantly, we demonstrate that our approach achieves better flow compliance than the baselines.

II. Related Work

The work presented in this paper is related to motion planning over vector fields, motion planning under sensing

[1] Mobile Robots and Olfaction Lab, AASS, Örebro University, Sweden. {chittaranjan.swaminathan, tomasz.kucner, martin.magnusson, achim.lilienthal}@oru.se

[2] Robert Bosch GmbH Corporate Research, Germany <Luigi.Palmieri@de.bosch.com>

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732737 (ILIAD).

978-1-5386-8093-3/18/$31.00 ©2018 IEEE

and motion uncertainty, and human aware motion planning. In this section we present works from these fields and compare them to our approach.

Kularatne et al. [4] use a graph-search based optimization technique to plan energy-efficient paths over vector fields. Another approach for planning over unimodal fields is the Vector Field RRT algorithm proposed by Ko et al. [5]. VF-RRT utilizes the *upstream criterion* to find trajectories that require least control effort for the robot. Existing approaches considering motion planning over vector fields assume unimodal characteristic of the vector field. In contrast, our work utilizes a more general multi-modal representation, i.e., more than one velocity vector can be associated to each point.

Melchoir et al. [6] propose a method called pRRT, to address the problem of planning under motion uncertainty. pRRT incorporates a particle-based extension technique that accounts for uncertainty. They use rejection sampling to bias the sampling step of RRT using the probability of success of a particular node or path. In our work, we use a similar sampling algorithm based on rejection sampling to drive RRT* to expand into regions with less motion. The RRBT algorithm developed by Bry et al. [7] addresses motion planning under sensing uncertainty. In their work, search is performed over a graph, where each vertex has an associated state and belief, to find the path with the lowest probability of collision. These works provide insight into how the probability of success might be used with the RRT planner to generate safer paths. However, they differ significantly from ours in that they do not explicity account for dynamics in the environment or the related uncertainty.

The CLiFF-RRT* algorithm by Palmieri et al. [3], is a modified RRT* algorithm that also uses CLiFF-maps. In particular, the cost function used is a modified upstream criterion similar to the one used by Ko et al [5]. However, the cost function of CLiFF-RRT* does not exploit all the information available in the CLiFF-map representation. The key conceptual differences between CLiFF-RRT* and our approach is discussed in Sec. IV-D.

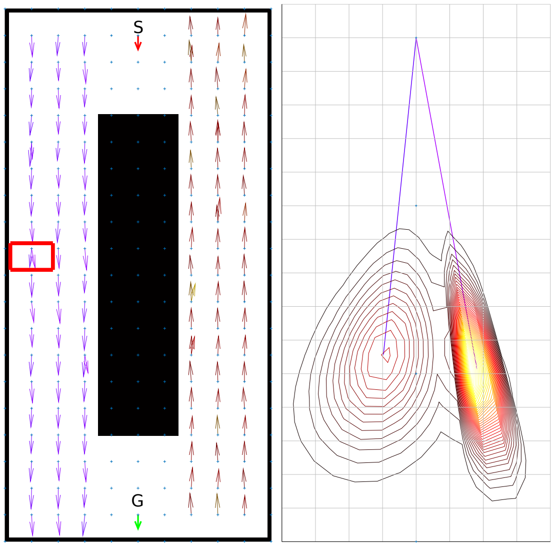
III. CLiFF-map representation

Kucner et al. [2] propose a method to model velocity observations independently from the underlying physical processes. Velocity, $\mathbf{V} = \begin{bmatrix}\theta & \rho\end{bmatrix}^T$, is represented using heading, $\theta \in [0, 2\pi)$, and speed, $\rho \in \mathbb{R}^+$, in a joint distribution.

CLiFF-map uses a mixture of Semi-Wrapped Normal Distributions (SWND) to model the Probability Density Function (PDF) of the velocity. A bivariate SWND has one dimension (speed, ρ) along the height of a cylinder and the other (heading, θ), wrapped along its circumference. An SWND with mean, $\boldsymbol{\mu}_{2\times1}$, and covariance matrix, $\boldsymbol{\Sigma}_{2\times2}$, is given by:

$$\mathcal{N}^{SW}_{\boldsymbol{\Sigma},\boldsymbol{\mu}}(\mathbf{V}) = \sum_{k\in\mathbb{Z}} \mathcal{N}_{\boldsymbol{\Sigma},\boldsymbol{\mu}}\left(\begin{bmatrix}\theta\\ \rho\end{bmatrix} + 2\pi\begin{bmatrix}k\\ 0\end{bmatrix}\right) \tag{1}$$

where k is the winding number. Although $k \in \mathbb{Z}$, according to Mardia and Jupp [8], in practice, taking $k = -1, 0, 1$ results in an adequate approximation of the PDF.



(a) An occupancy map overlaid with a CLiFF-map showing the means.

(b) A zoomed in image showing mean velocities for location (0.5,5.5) overlaid with the corresponding polar plot of the PDF of the velocity distribution.

Fig. 2: A CLiFF-map organized on a grid with cell-size of 0.5 m. Nearby distributions have been removed in the zoomed-in image for clarity. Start (S) and goal (G) poses used in experiments are indicated.

A Semi-Wrapped Gaussian Mixture Model (SWGMM) is employed to preserve the multi-modal nature of the flow. An SWGMM is a PDF represented as a combination of J SWNDs:

$$p(\mathbf{V}|\boldsymbol{\xi}) = \sum_{j=1}^{J} \pi_j \mathcal{N}^{SW}_{\boldsymbol{\Sigma}_j,\boldsymbol{\mu}_j}(\mathbf{V}) \tag{2}$$

where $\boldsymbol{\xi} = \{\xi_j = (\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \pi_j) | j \in \mathbb{Z}^+\}$. That is, $\boldsymbol{\xi}$ is an SWGMM represented by its means $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_J$, covariances $\boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_J$ and mixing factors $\pi_1, ..., \pi_J$. In practice, the number of SWNDs is determined by mean-shift clustering. Note that the SWGMM is a convex combination of the constituent SWNDs. Thus, every mixing factor π_j statisfies $0 \leq \pi_j \leq 1$ and the sum of all mixing factors is one.

The *motion ratio*, q_s, is the ratio of duration during which motion was observed (T^m_s), to the observation duration (T^o_s) for location $\mathbf{l}_s$: $q_s = T^m_s / T^o_s$. A motion ratio of one indicates that every time this location was observed, motion was also observed whereas, a zero value indicates that motion was never observed in the location. The *observation ratio*, p_s, is the ratio of observation duration (T^o_s) for location $\mathbf{l}_s$ to the total observation duration for all locations (T^t): $p_s = T^o_s / T^t$. An observation ratio of one indicates that the location was observed for the entire duration of observation, whereas a value less than one indicates partial observability, and a value of zero indicates no observation. In summary, CLiFF-map can be denoted as:

$$\Xi = \{(\boldsymbol{\xi}_s, p_s, q_s, \mathbf{l}_s) \mid s \in \mathbb{Z}^+ \wedge \mathbf{l}_s \in \mathbb{R}^2\} \tag{3}$$

Fig. 2 shows an example of a grid-based CLiFF-map, where the distributions are located over a regular grid.

IV. Trajectory planning over CLiFF-maps

We chose to use our cost function with the RRT* algorithm due to its favorable properties of asymptotic optimality and probabilistic completeness. We also propose an algorithm to bias the sampling process of RRT*.

Let $\mathcal{X} \subset \mathbb{R}^n$ be the configuration space of the robot and $\mathbb{U} \subset \mathbb{R}^m$ be the control space, where n is the dimensionality of the configuration space, and m that of the control space. Let $\mathcal{X}_{obs}$ be the obstacle space and $\mathcal{X}_{free} \subset \mathcal{X} \setminus \mathcal{X}_{obs}$ be the free space. Let the start state be $\mathbf{x}_{start} \in \mathcal{X}_{free}$ and the goal region be $\mathcal{X}_{goal} \subset \mathcal{X}_{free}$. The RRT* algorithm creates a tree in $\mathcal{X}_{free}$ whose nodes correspond to states in $\mathcal{X}_{free}$ and edges correspond to trajectories between states. The RRT* algorithm aims to minimize the cost function by continuously improving the initial trajectory. We consider robots with a Dubins kinematics [9]. Hence, the configuration space, $\mathcal{X}$, is three dimensional and a state in $\mathcal{X}_{free}$ is represented by the x and y co-ordinates and the heading, θ of the vehicle.

A. Cost function

We will penalize trajectories where the robot velocities deviate from the underlying flow pattern. In other words, we design cost functions that penalize motions that do not conform to the underlying CLiFF-map distributions. In this section we use the following concepts: A trajectory consists of a list of trajectory points. Each trajectory point is composed of the robot's position, heading and speed (magnitude of the velocity vector). The cost of a trajectory is the sum of costs of its trajectory points.

Suppose $\boldsymbol{\sigma}(i)$ denotes a trajectory point. Let $\mathbf{V}_i$ be the vector composed of the heading and speed associated with the trajectory point. Suppose there are J_i SWNDs at the CLiFF-map location corresponding to the position of the trajectory point $\boldsymbol{\sigma}(i)$. The Mahalanobis distance associated with the trajectory point is then:

$$D_i = \sum_{j=1}^{J_i} \left(\sqrt{(\mathbf{V}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{V}_i - \boldsymbol{\mu}_j)}\right) \tag{4}$$

The Mahalanobis distance is higher for velocity instances that do not conform to the underlying distribution. Hence we propose to use it as a component of the cost function.

Now, suppose a trajectory denoted as ς_a^1 is composed of trajectory points $\boldsymbol{\sigma}(1)$ through $\boldsymbol{\sigma}(a)$. The proposed cost due to CLiFF-map for this trajectory is the sum of Mahalanobis distances over all trajectory points:

$$c_c(\varsigma_a^1) = \sum_{i=1}^{a} (p_i q_i D_i) \tag{5}$$

where, p_i and q_i are the observation and motion ratios, respectively, at the CLiFF-map location corresponding to the position of trajectory point, $\boldsymbol{\sigma}(i)$. If, for example, the motion ratio for a CLiFF-map location is zero, the cost due to CLiFF-map is also zero at this location.

The pq weighting in Eq. (5) would lead to lower costs for locations with less motion or fewer observations. Hence, a planner using this cost function will generate trajectories along regions with less motion (*congestion-avoiding* due to q) and fewer observations (*exploratory* due to p), while simulataneously trying to match the speed and orientation of the underlying flow (*flow-conforming* due to Mahalanobis distance). The choice of weighting can change the quality of trajectories produced while still preserving flow-conformity. This is discussed in Section. IV-B.

In order to penalize long paths, we use a distance metric by LaValle et al. [10]. The distance cost associated with the trajectory is:

$$c_d(\varsigma_a^1) = \sum_{i=1}^{a} ||\mathbf{x}_i - \mathbf{x}_{i-1}|| + |(1 - |\mathbf{q}_i \cdot \mathbf{q}_{i-1}|)| \tag{6}$$

where $\mathbf{x}_i$ and $\mathbf{q}_i$ are, respectively, the position vector and quaternion associated with the trajectory point $\boldsymbol{\sigma}(i)$. The first term inside the sum is the Euclidean distance. The quaternion dot product $\mathbf{q}_i \cdot \mathbf{q}_j$ yields the cosine of the angle between the two unit quaternions. Hence, the term $|(1 - |\mathbf{q}_i \cdot \mathbf{q}_{i-1}|)|$ results in trajectories with fewer turns.

The total cost of a trajectory to a node in the search can then be written as:

$$c_{tot}(\varsigma_a^1) = w_c c_c(\varsigma_a^1) + w_d c_d(\varsigma_a^1) \tag{7}$$

where w_d and w_c are weighting parameters for the costs due to distance and flow respectively. Having a low w_d and high w_c would make the planner prefer longer and flow-conforming motions compared to shorter and non-flow-conforming motions. The weighting parameters can also be used to tune the planner to adopt a *flow-ignorant* ($w_c = 0$) or *strictly flow-following* ($w_d = 0$) planning strategy.

B. Cost function characteristics

Our cost function produces an *exploratory, congestion-avoiding and flow-conforming* behaviour. We further note the following: Given two CLiFF-map regions, the cost function would lead to trajectories (1) along regions with lower product of motion and observation ratios; (2) so that the robot's speed and heading matches the CLiFF-map mean at each trajectory point; (3) along regions where the covariance is larger.

The *exploratory and congestion-avoiding* characteristics may not be preferred always. However, the expression in Eq. (5) can easily be tuned to yield different characteristics. For instance, consider the following expressions:

$$c_c(\varsigma_a^1) = \sum_{i=1}^{a} (q_i D_i) \tag{8}$$

$$c_c(\varsigma_a^1) = \sum_{i=1}^{a} (\frac{q_i}{p_i} D_i) \tag{9}$$

Eq. (8) assumes that the cost is independent of how long a location was observed (assumes observation ratio, $p = 1$). Hence, the behaviour of a robot utilizing this cost function can be described as *confident* since it assumes perfect knowledge of the flow. Eq. (9), on the other hand, would lead to a *cautious* behaviour since it would prefer regions that were observed a lot. The longer a location was observed, the smaller the cost.

C. Sampling function

We now propose a biased sampling function for use with sampling-based motion planners. We use rejection sampling from a biased distribution in order to guide the RRT*'s exploration of the configuration space. Our design of the sampling bias is directly motivated by our choice of cost function: we will use the motion and observation ratios in the sampling procedure so as to prefer an *exploratory and congestion-avoiding* behaviour that guides exploration towards regions with fewer observations or less motion (Eq. (5)), while also trying to closely match the modes of flow that have been learnt.

The pseudo-code for sampling is presented in Alg. 1. The biasing strategy is as follows: With a 5% probability a state is chosen from the goal region. Otherwise a state is sampled uniformly from the entire state space. With a 20% probability, this sample is unconditionally accepted. This number was arbitrarily picked to ensure probabilistic completeness. Otherwise, we do the following: if motion ratio, is low, we accept the sample, since either the cost is low (if observation ratio is also low) or we are confident that there is less motion in this region. If motion ratio is high, but observation ratio is low, we try to follow the flow. If both the observation and motion ratios are high, then we reject the sample (cost is likely to be higher).

Algorithm 1: Pseudo-code for sampling process.

```
Result: Newly sampled pose $\mathbf{x_{new}}$
Generate 3 random numbers $r1$, $r2$ and $r3$ from a uniform
  distribution;
if $r1 < 0.05$ then
    $\mathbf{x_{new}} \leftarrow UniformSample(\mathcal{X}_{goal})$;
    return $\mathbf{x_{new}}$;
$\mathbf{x_{new}} \leftarrow UniformSample(\mathcal{X}_{free})$;
if $r1 > 0.80$ then
    return $\mathbf{x_{new}}$;
while True do
    $q \leftarrow cliffmap(\mathbf{x_{new}}).q$
    $p \leftarrow cliffmap(\mathbf{x_{new}}).p$
    if $r2 > q$ then
        return $\mathbf{x_{new}}$;
    else
        if $r3 < pq$ then
            Reject the sample;
        else
            $\mathbf{x_{new}}.heading \leftarrow$
              $cliffmap(\mathbf{x_{new}}).heading$;
            return $\mathbf{x_{new}}$;
```

D. Comparison between DTC-RRT* and CLiFF-RRT*

In this section we outline key differences between CLiFF-RRT* by Palmieri et al. [3] and the DTC-RRT* algorithm proposed in this paper. The cost function used in CLiFF-RRT* is a combination of the distance metric in Eq. (6) and the upstream criterion. Similar to Eq. (4), the *upstream criterion* or *upstream cost* associated with a trajectory point $\boldsymbol{\sigma}(i)$ can be written as:

$$U_i = \sum_{j=1}^{J_i} (||\vec{\boldsymbol{\mu}}_j|| - \langle \vec{\boldsymbol{\mu}}_j, \hat{\mathbf{x}}_i \rangle) \tag{10}$$

where $\mathbf{x}_i$ is the position vector associated with the trajectory point $\boldsymbol{\sigma}(i)$, and $\hat{\mathbf{x}}_i$ is the unit tangent vector along the trajectory direction at the trajectory point. Note that the CLiFF-map means used in Eq. (10) are converted to Cartesian coordinates and hence, the arrow notation. The entire cost function is:

$$c_{dU}(\boldsymbol{\varsigma}_a^1) = U(\boldsymbol{\varsigma}_a^1) + c_d(\boldsymbol{\varsigma}_a^1) \tag{11}$$

where c_d is the distance metric from Eq. (6) and $U(\boldsymbol{\varsigma}_a^1)$ is the sum of all upstream criteria over all trajectory points in the trajectory, $\boldsymbol{\varsigma}_a^1$.

Comparing our cost function (Eq. (7)) with the one used in CLiFF-RRT* (Eq. (11)), we see that the latter lacks the weighting parameters w_d and w_c. Hence it lacks the ability to tune the planner to adopt either a *strictly flow-following*, a *flow-ignorant* or any other strategy inbetween the two by adjusting the relative values of w_d and w_c.

The upstream criterion used in CLiFF-RRT* does not consider the mean speed of the distribution, only the mean direction. CLiFF-RRT* does not respond to information about uncertainty. However, due to the use of Mahalanobis distance, the DTC cost function explicitly considers the entire mean and covariance of the distribution.

In addition to uniform sampling, CLiFF-RRT* uses a set of CLiFF-map mixtures to sample a new state. In contrast, our method utilizes p and q to sample a new state. The difference is qualitative: while CLiFF-RRT* biases search towards a set of CLiFF-map mixtures (into the flow), our algorithm biases search towards regions with low motion ratio, q (away from the flow).

V. Experiments and Discussion

In this section we describe the experimental setup and discuss the results of the evaluation of our algorithm. We ran DTC-RRT* and two base line planners - uninformed RRT* and Upstream-RRT*, the main focus being on the cost function. The Upstream-RRT* is CLiFF-RRT* with uniform sampling (without the sampling bias).

A. Evaluation setup

For the qualitative experiments we compare the trajectories generated by DTC-RRT* with those generated by Upstream-RRT*. We will also verify our expectations from Section IV-B about the characteristics of trajectories obtained from

DTC-RRT* algorithm. In the quantitative experiments, we measure the planning time, distance costs (c_d), upstream costs (U) and DTC costs (c_{tot}) and perform a comparison between uninformed RRT*, Upstream-RRT*, unbiased DTC-RRT*, DTC-RRT* with sampling bias function. All the aforementioned quantities are the latest solutions within a planning time limit of sixty seconds. In particular, planning time here refers to the time at which the latest solution was obtained. That is, after this time no improvements were made by the RRT* algorithm to the solution trajectory.

Fig. 2 shows the occupancy grid of the environment used for evaluation overlaid with a CLiFF-map grid. Start and goal positions used in the evaluation are marked as *S* and *G* respectively. We built various CLiFF-maps, each with different parameters on either side of the central obstacle on the map in order to see which side the planner picks while reaching from start to goal. We built the CLiFF-maps using virtual velocity observations generated as follows: we use two rectangular regions on either side of the central obstacle in the map (Fig. 2) and specify a set of parameters for each rectangular area: (1) the mean and variance of the speed and orientation of the observations, and (2) the number of observation samples in each rectangular area.

Different virtual observations for eight maps are generated according to the parameters listed in Table I chosen to represent key differences a planning algorithm should consider. The eight maps used can be described in words as follows: (1) *DiffDirections* map has two flows with opposite mean directions; (2) *DiffSpeeds* map has two flows with different mean speeds; (3) *DiffQ* map has one flow on the left side but no flow on the right, hence the two regions have different motion ratios; (4) *DiffVarSpeed* has two flows with different variance in speed; (5) *DiffVarDir* has two flows with different variance in orientation; (6) *DiffIntensity* has two flows where the corresponding regions have different motion ratios; (7) *DiffPQ1* and (8) *DiffPQ2* have two flows with different motion and observation ratios. In *DiffPQ1*, the average pq value on the left hand size was about 0.17, whereas on the right it was 0.21. In *DiffPQ2*, the average pq value on the left was 0.32, whereas on the right it was 0.11.

B. Results and observations

Table II presents the quantitative results. Each algorithm was run 25 times on each of the maps. Success is computed as the percentage of trials in which a solution was found under a time limit of sixty seconds. The results can be summarized as follows:

(a) Unbiased DTC-RRT* takes much longer to improve solutions compared to RRT* and Upstream-RRT*.

(b) DTC-RRT* with biased sampling takes at most the same amount of time as the unbiased version (Fig. 3). Also, the median planning times are very close to the planning times of RRT* and Upstream-RRT*.

(c) DTC-RRT* yields less expensive solutions than baselines in terms of all three costs. In fact DTC-RRT* with a sampling bias yields solutions with lower upstream cost compared to Upstream-RRT*. Minimizing the Mahalanobis distance should automatically minimize upstream cost because, both the vehicle speed and heading at each trajectory point will be closer to the mean of underlying flows when the Mahalanobis distance is minimized. Also, since DTC-RRT* uses biased sampling, it better minimizes the Mahalanobis distance compared to the unbiased variant of DTC-RRT*. This is possibly why it better minimizes the upstream criterion.

The qualitative results are presented in the form of heatmaps of solutions obtained using DTC-RRT* and Upstream-RRT* planners in Fig. 4 and Fig. 5. The heatmaps were constructed by accumulating the latest solutions after sixty seconds and computing the number of trajectory points that fall into each cell on the grid map. The following qualitative observations can be made from the heatmap:

(i) DTC-RRT* achieves better *flow conformity* compared to Upstream-RRT*. It prefers regions where the vehicle speed and direction matches those of the underlying distribution (Fig. 4b). Since the Dubins vehicle has a constant velocity of one metre per second, DTC-RRT* picks trajectories on the left side where the speeds are closer to one metre per second (see Table. I). The cost function of Upstream-RRT* does not consider the mean speed.

(ii) DTC-RRT* uses motion ratio information to find trajectories along regions with no motion (Fig. 4c), or less motion (Fig. 4d) as compared to Upstream-RRT*.

(iii) In *DiffPQ2*, DTC-RRT* finds trajectories predominantly along regions with lower product of motion and observation ratios, (Fig. 4h). In *DiffPQ1*, pq on the left and right side are almost equal. Therefore, the result from DTC-RRT* (Fig. 4g) is very similar to the corresponding result from Upstream-RRT* (Fig. 5g), which does not use p or q.

(iv) DTC-RRT* finds trajectories predominantly along regions with greater covariance (Figures 4e and 4f) as compared to Upstream-RRT*.

VI. Conclusions and Future Work

In this paper we address the problem of flow-aware trajectory planning that considers uncertainty due to partial observability and statistical variance. We have proposed generalized cost and biasing functions that utilize all the information available in the CLiFF-map model. We have implemented one flavour of the proposed cost and biasing functions and compared its evaluation results to those obtained using uninformed RRT* and the Upstream-RRT* algorithm. The trajectories resulting from our algorithm are quantitatively better: they respect the underlying flow better than the baselines. The results also show that the chosen flavour of the cost and biasing functions would yield trajectories with the *exploratory, congestion-avoiding and flow-conforming* characteristics.

However, we could make adjustments to the cost and biasing functions so as to achieve qualitatively different trajectories. For instance, we might want the robot to adopt a more *cautious* or *confident* strategy (Sec. IV-B). In the future we would like to demonstrate and evaluate the effects of the cost functions and their combinations. The sampling strategy

TABLE I: Different parameters used to generate the input data. Speeds are in meters per second, angles in degrees.

Map ID	**Speed (left)**	**Heading (left)**	**Speed (right)**	**Heading (right)**	**Samples (left)**	**Samples (right)**
DiffDirections	1.0 ± 0.45	-90 ± 3.35	1.0 ± 0.45	90 ± 3.35	10k	10k
DiffSpeeds	1.0 ± 0.45	-90 ± 3.35	0.5 ± 0.45	-90 ± 3.35	10k	10k
DiffQ	1.0 ± 0.45	-90 ± 3.35	-	-	10k	0
DiffVarSpeed	1.0 ± 0.32	-90 ± 3.35	1.0 ± 0.78	-90 ± 3.35	10k	10k
DiffVarDir	1.0 ± 0.45	-90 ± 3.35	1.0 ± 0.45	-90 ± 4.75	10k	10k
DiffIntensity	1.0 ± 0.45	-90 ± 3.35	1.0 ± 0.45	-90 ± 3.35	20k	10k
DiffPQ1-2	1.0 ± 0.45	-90 ± 3.35	1.0 ± 0.45	-90 ± 3.35	20k	20k

TABLE II: Results from evaluation.

	Planning time	**DTC cost (c_{tot})**	**Upstream cost (U)**	**Distance cost (c_d)**	**Success**
			Uninformed RRT*		
DiffDirections	9.92 ± 12.74	7793.50 ± 4064.84	53.33 ± 41.2	33.48 ± 1.89	90%
DiffSpeeds	7.50 ± 10.62	7782.19 ± 3320.10	4.56 ± 2.62	33.65 ± 1.63	
DiffQ	8.60 ± 8.35	4422.69 ± 3881.85	4.44 ± 3.45	33.57 ± 1.30	
DiffVarSpeed	7.24 ± 8.64	7768.74 ± 6740.81	3.83 ± 2.65	33.30 ± 1.17	
DiffVarDir	6.66 ± 10.10	7470.19 ± 4566.96	5.13 ± 4.30	33.69 ± 1.54	
DiffIntensity	7.65 ± 8.81	5122.64 ± 3820.58	3.90 ± 2.94	33.34 ± 1.55	
DiffPQ1	6.86 ± 10.21	132.14 ± 437.01	3.67 ± 2.23	33.47 ± 1.65	
DiffPQ2	8.50 ± 9.43	804.95 ± 2232.86	9.81 ± 17.27	34.33 ± 1.83	
			Upstream-RRT*		
DiffDirections	10.74 ± 12.44	7408.48 ± 4157.70	4.43 ± 2.40	34.83 ± 1.87	89.5%
DiffSpeeds	7.80 ± 7.95	6904.24 ± 5197.57	**1.46** ± 2.25	34.95 ± 1.87	
DiffQ	4.03 ± 5.24	893.27 ± 1741.86	0.68 ± 2.38	34.60 ± 1.83	
DiffVarSpeed	8.60 ± 7.40	4898.31 ± 7498.97	2.41 ± 1.54	34.72 ± 1.60	
DiffVarDir	9.80 ± 11.15	4301.20 ± 3682.80	2.41 ± 1.52	35.04 ± 1.78	
DiffIntensity	8.51 ± 7.57	3082.43 ± 2618.13	2.50 ± 1.36	34.14 ± 1.42	
DiffPQ1	5.77 ± 5.36	124.13 ± 416.92	2.34 ± 1.48	34.54 ± 1.66	
DiffPQ2	8.05 ± 10.18	1156.52 ± 1640.28	2.12 ± 1.85	34.46 ± 1.89	
			DTC-RRT* with uniform sampling		
DiffDirections	21.86 ± 14.06	49.14 ± 3.64	5.34 ± 3.13	38.22 ± 2.75	92.5%
DiffSpeeds	18.80 ± 16.19	47.64 ± 2.28	3.81 ± 3.77	36.27 ± 1.94	
DiffQ	8.70 ± 8.13	35.47 ± 2.39	**0.00** ± 0.00	35.47 ± 2.39	
DiffVarSpeed	22.56 ± 13.36	42.59 ± 1.69	1.87 ± 1.29	36.76 ± 1.44	
DiffVarDir	19.30 ± 15.85	41.67 ± 2.81	2.80 ± 2.11	36.33 ± 1.97	
DiffIntensity	20.17 ± 14.59	40.77 ± 3.13	2.50 ± 2.13	36.60 ± 2.44	
DiffPQ1	7.95 ± 9.79	36.91 ± 1.81	2.62 ± 2.21	34.35 ± 1.92	
DiffPQ2	13.27 ± 11.54	36.53 ± 1.90	2.39 ± 2.07	34.68 ± 1.83	
			DownTheCLiFF RRT*		
DiffDirections	13.02 ± 14.47	44.99 ± 3.20	**3.81** ± 2.54	36.87 ± 2.15	87.5%
DiffSpeeds	9.88 ± 11.80	41.29 ± 2.27	2.69 ± 2.07	35.09 ± 2.08	
DiffQ	5.83 ± 5.42	34.27 ± 1.55	0.03 ± 1.53	34.16 ± 1.61	
DiffVarSpeed	21.22 ± 14.38	38.58 ± 1.48	**0.97** ± 1.72	34.76 ± 1.93	
DiffVarDir	14.61 ± 14.97	37.55 ± 1.13	**1.68** ± 1.27	34.03 ± 1.18	
DiffIntensity	14.79 ± 15.04	36.85 ± 1.84	**1.84** ± 1.82	33.80 ± 1.95	
DiffPQ1	13.62 ± 14.61	34.84 ± 1.05	**1.59** ± 1.09	**32.95** ± 1.14	
DiffPQ2	15.84 ± 16.37	34.70 ± 1.29	**1.65** ± 1.81	33.39 ± 1.30	

currently used relies on rejection sampling. We would also like to work on more sophisticated biasing strategies. We also hope to utilize vehicle models with different kinodynamic constraints.

References

[1] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore, *et al.*, "Spencer: A socially aware service robot for passenger guidance and help in busy airports," in *Field and service robotics*, pp. 607–622, Springer, 2016.

[2] T. P. Kucner, M. Magnusson, E. Schaffernicht, V. H. Bennetts, and A. J. Lilienthal, "Enabling flow awareness for mobile robots in partially observable environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1093–1100, 2017.

[3] L. Palmieri, T. P. Kucner, M. Magnusson, A. J. Lilienthal, and K. O. Arras, "Kinodynamic motion planning on gaussian mixture fields," in *IEEE International Conference on Robotics and Automation (ICRA), 2017*, pp. 6176–6181, IEEE, 2017.

[4] D. Kularatne, S. Bhattacharya, and M. A. Hsieh, "Time and energy optimal path planning in general flows.," in *Robotics: Science and Systems*, 2016.

[5] I. Ko, B. Kim, and F. C. Park, "Randomized path planning on vector fields," *The International Journal of Robotics Research*, vol. 33, no. 13, pp. 1664–1682, 2014.

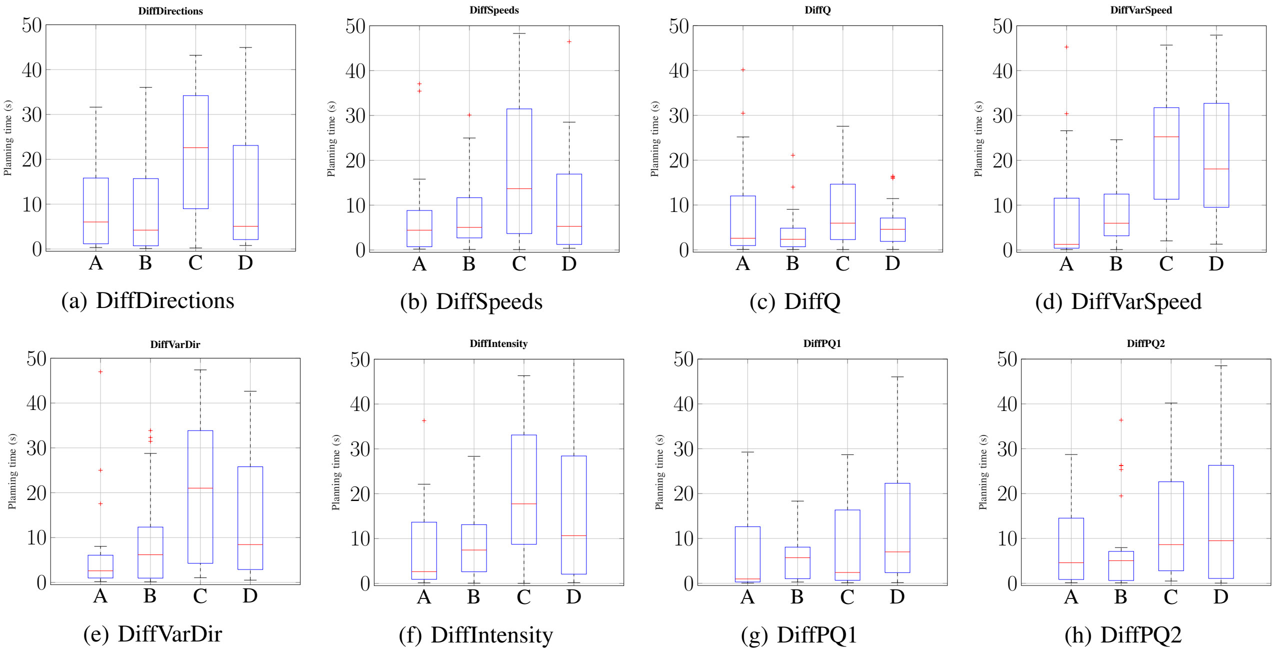
[6] N. A. Melchior and R. Simmons, "Particle RRT for path planning with uncertainty," in *IEEE International Conference on Robotics and Automation, 2007*, pp. 1617–1624, IEEE, 2007.

[7] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *IEEE International Conference on Robotics and Automation (ICRA), 2011*, pp. 723–730, IEEE, 2011.

[8] P. E. Jupp, *Directional statistics*. Wiley Online Library, 2001.

[9] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

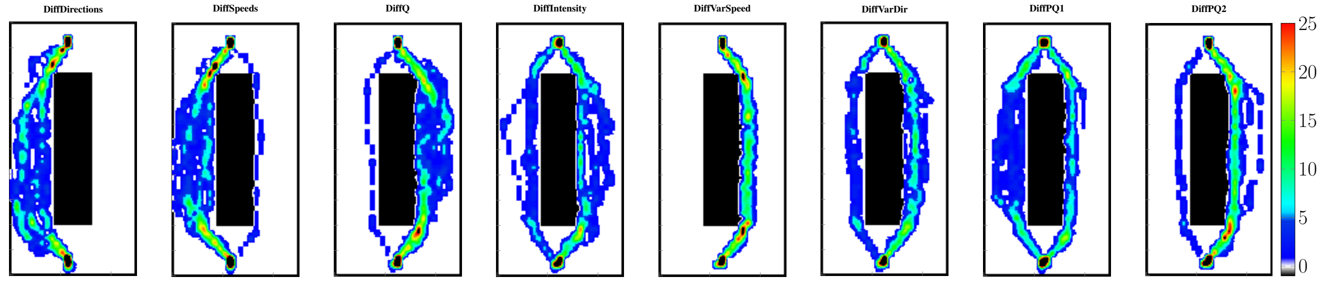
[10] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.



(a) DiffDirections (b) DiffSpeeds (c) DiffQ (d) DiffVarSpeed

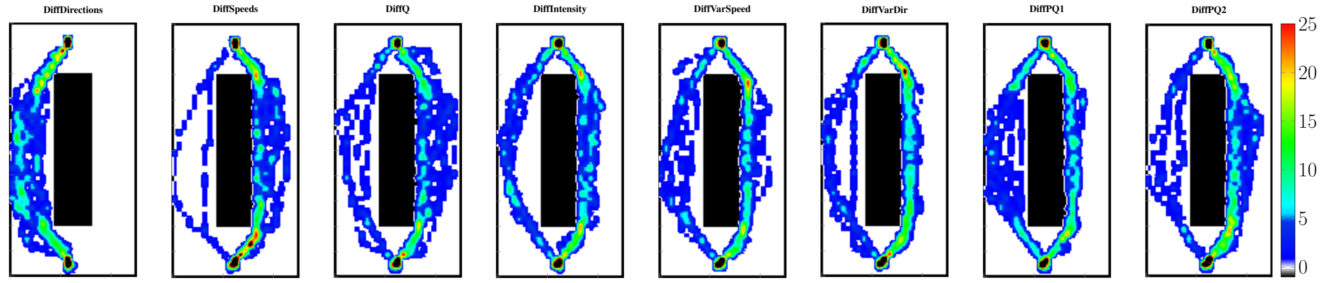
(e) DiffVarDir (f) DiffIntensity (g) DiffPQ1 (h) DiffPQ2

Fig. 3: Box plot of time for latest trajectory within a time limit sixty seconds from (A) RRT*, (B) Upstream-RRT*, (C) Unbiased DTC-RRT*, (D) DTC-RRT*.



(a) DiffDirections (b) DiffSpeeds (c) DiffQ (d) DiffIntensity (e) DiffVarSpeed (f) DiffVarDir (g) DiffPQ1 (h) DiffPQ2

Fig. 4: Heat maps for trajectories generated by DTC-RRT*. Colors indicate number of trajectory points at a cell. Black denotes an obstacle, White denotes free space.



(a) DiffDirections (b) DiffSpeeds (c) DiffQ (d) DiffIntensity (e) DiffVarSpeed (f) DiffVarDir (g) DiffPQ1 (h) DiffPQ2

Fig. 5: Heat maps for trajectories generated by Upstream-RRT*. Colors indicate number of trajectory points at a cell. Black denotes an obstacle, White denotes free space.