Autonomous Kinodynamic Path Planning for Following and Tracking Vehicles

Jonatan Scharff Willners*†, Yvan R. Petillot†, Pedro Patron* and Daniel Gonzalez-Adell*†

*SeeByte, Edinburgh, UK, {jonatan.willners, pedro.patron}@seebyte.com
†School of Electrical Engineering, Heriot-Watt University, Edinburgh, UK, {js100, y.r.petillot, dg36}@hw.ac.uk

***Abstract*—This paper presents a novel approach to planning vehicle paths under kinodynamic constraints in a leader follower scenario where the follower vehicle has to track and follow the leader. This problem is important in the maritime domain where Autonomous Underwater Vehicles (AUVs) can greatly benefit from an Autonomous Surface Vehicle (ASV) acting as a Communication Relay (CR) and/or a Navigational Aid (NA), typically using acoustic communication. The proposed approach is an extension of Hybrid-A* (HA*), a hybrid version of A* which enables the derivation of paths that are obstacle free and feasible by the vehicle. The proposed algorithm finds a solution, if it exists, for scenarios where the leader and the follower operate under the same kinematic constraints as well as when they differ. Various simulations using multiple configurations and scenarios are presented to validate the approach. Whilst the work presented here has a focus on the maritime environment, the algorithm is applicable to other domains.**

***Keywords*—ASV, AUV, Path Planning, Kinodynamics, Cooperative robotics.**

I. INTRODUCTION

In a maritime environment, having a surface vehicle in close proximity with a submerged vehicle can offer great benefits. It can act as both a CR and as a NA, typically over acoustic communication. This is as electromagnetic waves do not propagate subsea, hence limiting communications and access to Global Positioning System (GPS). Acoustic communications are both low bandwidth and can be unreliable due to the nature of the acoustic channel in water. These two constraints are the main reasons why an AUV periodically needs to surface, either to regain certainty in its position and/or to access other means of communication, for example to exchange data with operators in a Command and Control Centre (CCC). With a surface vehicle, such as an ASV planning its path to be in close proximity to the AUV, it can act as a CR and as an acoustic beacon used for localization. As a CR, in extends the communication range for the AUV to close to real-time [1] to other platforms above surface without the AUV having to surface. As an acoustic beacon, the ASV can through acoustic communication, aid the AUV with localization by Ultra-short Base-line (USBL) or Moving Long Baseline (MLBL) [2], [3], [4]. An example is shown in figure 1. Here an ASV is updating an AUV through USBL while the latter is submerged. Vehicles of different types are usually under different motion constraints. For a maritime environment it is not uncommon that the surface vehicle is driven by petrol engines. These engines might force the vehicle to drive at a minimum speed that is greater than the AUV's maximum operating speed. Therefore, having them execute the same path might not be feasible and a different type of planner is required. This paper presents a planner that deals with both the scenario where the vehicle can drive at the same speed, as well as when they are forced to drive at different ones. The planner finds a drivable and collision free path for one vehicle to track and follow another.

The rest of the paper is organized as the following; in section II, a brief background on other work performed where a surface vehicle is tracking another vehicle is presented. In section III, the integration of kinodynamic constraints is presented, followed by the proposed algorithm. This is followed by the results in IV and a conclusion and the future of this work in V.

Fig. 1: Example of cooperative mission between a surface and subsurface vehicle. ASV's C-Worker is equipped with an USBL to support NOC's Autosub Long Range performing a survey.

II. BACKGROUND

The scenario of having an ASV tracking and/or following an AUV has already been studied. However, studies to date have focused scenarios where the vehicles have similar kinodynamic constraints where the ASV is typically a small electric vehicle. Such vehicles can operate accurately at low speeds (1-2 knots) compatible with typical AUV operations. In most cases, the algorithms developed to date aim at developing a control strategy which keeps the ASC above the AUV during mission. Melo and Matos [5], [6] use a Proportional Integral

(PI) controller to control the heading and velocity of the ASV to follow the same trajectory as the AUV. In [7], Bibuli *et al.* extend a nonlinear Lyapunov-based control with a virtual target to achieve a similar objective. This is done in two scenarios: an ASV following an ASV with an offset and ASV following an AUV. It is mentioned by the authors that freely driving vehicles can be hard as they might have high curvature segments in cases with slow dynamics. As such, narrow and tricky maneuvers could lead to unpredictable motions. Such movements could be potentially dangerous depending on the environment. This is a good argument for the importance of taking the kinodyanmic constraints of a platform into consideration when planning a path. In a survey performed by Kumru *et al.* [8], four different approaches for path tracking are compared; PID, pole-placement, feedback linearization and sliding mode control. The study includes disturbances (as small waves). The ASV is following the path of the AUV, which is know pre-mission. It follows the path with what seems to be no error compared to the AUV, except for when the AUV changes direction. Based on this, it occurs that the leader and follower suffer from similar kinematic constraints.

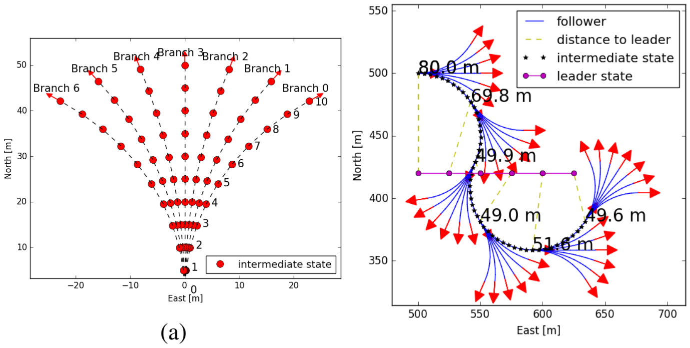
The work up to date tends to focus on a follower being able to precisely follow the leader. This indicates that the work is most likely not taking different kinematic constraints into consideration. In robotics it is necessary to operate in a safe and feasible manner. Based on this, the suggested path planner in this paper would contribute to a more reliable autonomous system.

III. KINODYNAMIC PATH PLANNING FOR VEHICLE FOLLOWING

Kinodynamic path planning is the art of planning a path that is both collision-free and drivable. That is, that the resulting path should not collide with any obstacles known at the stage of planning and that the path is constructed so that it is within possible motions for the robot.

The proposed planner is a hybrid between sampled based planning such as Rapid-exploring Random Trees (RRTs) [9] and heuristic-based planners such as A* [10] and HA* [11]. From an initial state in the Configuration Space (C-Space), a set of reachable states are generated by sampling the set of all reachable states under kinodynamic constraints (expansion step). As in A* the list of viable nodes are kept and the most promising one is selected for further expansion. The process is repeated until the termination condition is reached or no more nodes are available for expansion (no solution). The C-Space is the space that represents all possible states for the robot. In this work the dimensions for the C-Space are respectively in C^5 and C^3 for the AUV *(North, East, Depth, Heading, Pitch)* and the ASV *(North, East, Heading)*.

Expansion step: The expansion of nodes is performed by sampling the set of reachable positions in C-space using the kinodynamic constraints of the vehicle. We assume the kinodynamic constraints fixed for a specific vehicle and the set of possible positions can therefore be pre-calculated once to generate what we call a pattern. This pattern is a description of the possible nodes that can be reached under the vehicle's kinematic constraints. The pattern, as seen in figure 2, consists of i different branches. A branch consists of j intermediate states which are used for collision detection along the branch. Each branch is a description of a continuous motion where a constant force is applied to the actuators of the vehicle. Whilst this approach subsamples the search space and might lead to suboptimal solutions, the computational gain of using fixed patterns enables real time implementation which is a major advantage of our approach.



(a)

(b)

Fig. 2: (a): Pre-calculated pattern with with possible turning radii of 50, 76 and 152 m, 5m/s for 10s (10 intermediate steps). (b): Follower expanding using pattern in (a) multiple times to the state with the current lowest average distance to leader. The average distance can be seen as the number next to the chosen node. The AUV it is following has a linear speed of 2m/s.

When a node is expanded, the pattern is transformed to the state of the expanding node. This is achieved by the transformation matrix (T) seen in equation (1), where h is the heading and p is the pitch of the state currently being expanded. The new position of the pattern is given by equation 2.

$$T = \begin{bmatrix} \cos(h) * \cos(p) & -\sin(h) & \cos(h) * \sin(p) & x \\ \sin(p) * \cos(h) & \cos(h) & \sin(p) * \sin(h) & y \\ -\sin(p) & -\sin(h) & \cos(h) * \sin(p) & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$P_{new} = T * P^{\top} \tag{2}$$

When a state is expanded, each branch is expanded separately. For each branch the intermediate states are transformed and their validity is checked against specified conditions. These could for instance be collision detection with obstacles, exclusion zones, specific vehicle limitations or Automatic Identification System (AIS) data. If an intermediate state is not accepted, the rest of the branch is unreachable and the extension of that branch is discarded. If the last intermediate state of a branch is accepted, that node is added as a new viable node for the planner. For collision detection, Flexible Collision Library (FCL) [12] is used. The environment is represented as an OctoMap [13].

Termination step: The termination condition for classic path planning is to reach a specific goal location. In our case, there is no fixed goal location but rather an estimated leader trajectory over a specific time window. The planner aims at minimizing the average distance between the path generated

Kinodynamic Path Follower - Results ASV following AUV			
Turning radii (m)	number of branches per pattern	possible speeds for follower (m/s)	Avg. dist between leader and follower (m)
$25, 32, 50, 101$	9	0, 1, 2, 3, 4, 5	5.5
$25, 50$	5	2, 2.5, 5	7.20
$25, 50$	5	0, 5	17
$25, 50$	5	2.5, 5	19.14
$25, 50$	5	2.5	24.18
$25, 50$	5	2	32.23
$25, 50$	5	5	33.58

TABLE I: Path following comparison - different motion constraints. The leader performs the lawnmower shown in figure 4a with a constant speed of 2m/s.

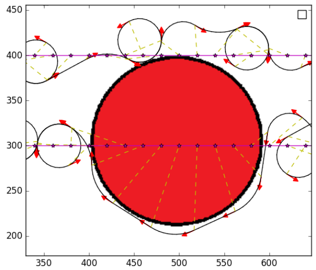


Fig. 3: Surface vehicle planning a path that avoids an obstacle at the surface (red circle) whilst following an underwater vehicle (dotted line). The ASV has a turning radii of 25 or 50 m and a linear speed of 5m/s whilst the AUV has a linear speed of 2m/s.

and the AUV's position for each time-step in the time window. The vehicle to follow is acting as a leader and updates the follower with its current state (position, speed and direction) referred to as $X(0)$ at the beginning of each time window. With this information, the follower can estimate the predicted position of the Target/leader ($\hat{X}(t)$) at time t as in (3). At each step of the expansion, the follower knows its own position $P(t)$ along the current best trajectory known position at time t and therefore the average distance along the trajectory can be estimated at all times. In the planner, each valid node (i.e. that can still be expanded) knows how many discrete time-steps n it is from the initial position at the start of the time window. The function $f(X)$, seen in equation (5), can therefore be used to evaluate and prioritize the node against other possible nodes.

$$\hat{X}(t) = \hat{X}(0) + \overrightarrow{v} * t \tag{3}$$

$$\begin{aligned} \hat{X}(0) = &\begin{bmatrix} X_{0_x} & X_{0_y} & X_{0_z} \end{bmatrix} = \\ &\begin{bmatrix} X_{0_x} & X_{0_y} & X_{0_z} \end{bmatrix} + (\overrightarrow{v} * 0) \end{aligned} \tag{4}$$

$$f(X) = \frac{\sum_{i=0}^{n} \left\| (P(i) - \hat{X}(i)) \right\|}{n} \tag{5}$$

The pseudo code for the proposed algorithm can be seen seen in Algorithm 1 and takes the following inputs:
$state$: The initial state of the vehicle. $(P(0))$
$pattern$: A pre-calculated pattern of feasible motions.
$environment$: Map of the current known environment.
$targetStart$: Start position of the leader. $(X(0))$
$speedVector$: Speed vector of the leader . $(\overrightarrow{v})$
$time$: the size of the time window. (t)

An example of a path found by the algorithm can be seen in figure 3. In this scenario, the ASV has a turning radii of 25 or 50 m and a linear speed of 5m/s and the AUV has a constant speed of 2m/s units per time-step. There is an obstacle on the surface for the ASV depicted by the red circle. The ASV successfully plans its path around the obstacle while trying to minimize its distance to the AUV.

IV. RESULTS

The algorithm has been tested in various scenarios and under different motion constraints for two cases; an ASV tracking and following an AUV and for an AUV tracking and following an AUV.

We first evaluate the performances of the algorithm under different kinodynamic constraints. The evaluation criteria are the average and maximum distance to the tracked AUV. The AUV is performing a lawnmower pattern seen in figure 4c(a). The AUV has a constant speed of 2m/s. In this scenario there are no obstacles but similar behavior is observed when there are.

The results of the evaluation are presented in table I. The vehicle model in this case is of a boat equipped with a rudder and we have used a simple motion model to convert rudder angle to turning radius. In the first column, the resulting turning radii being used for the pattern are displayed. As expected, the more freedom the follower has in its motion, the lower the average distance between the leader and the follower will be. However, the algorithm is able to generate a feasible path in real time even when the two vehicles have very different motion capabilities, which paves the way for on-board implementation on real platforms.

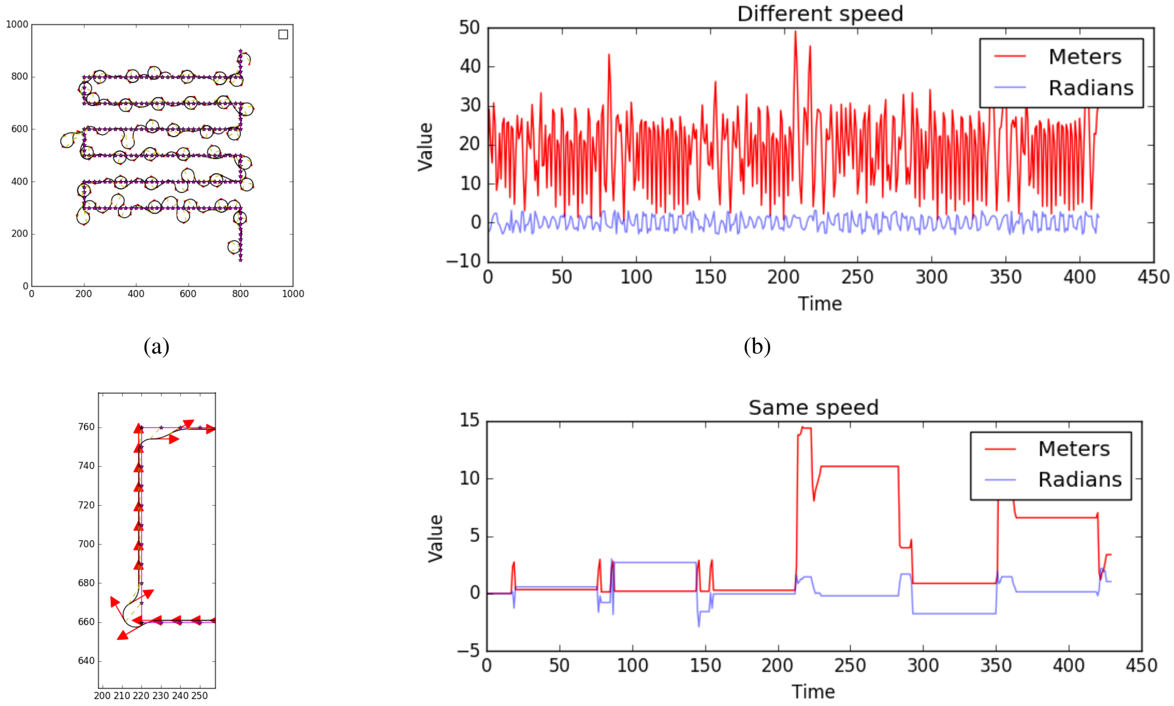
We also tested the case where the ASV is either static or circling around the center of the survey with a constant speed of 5m/s. The resulting average distance can be seen in figure 6a. An example on how these patterns would look is shown in figure 6b. From the graph in figure 6a, it can be seen that the lowest average distance for this approach is at a circle with a radius of 93 units resulting in a average distance of 205.8 units, over 6 times larger than the highest scenario in table I.

A. Benefits for Localisation

As seen in figure 4b, a vehicle that operates at a different speed than the vehicle it follows produces relative trajectories between the vehicles that vary in range and orientation over time. Moreover, these variations can be controlled by altering the turning radius of the vehicle. Such trajectories have been proven to enable robust and accurate localisation of the leader using either MLBL or trilateration[14].

B. Results in 3-Dimensional (3D)

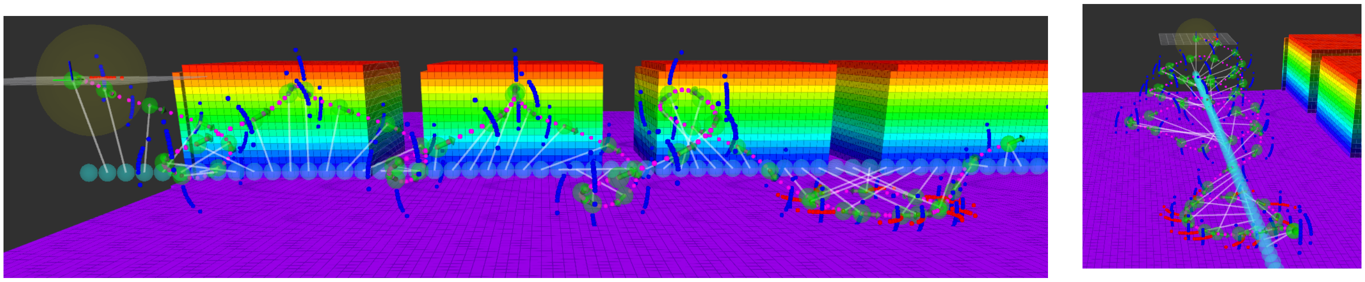
The algorithm has been tested in the scenario where an AUV is tracking and following another AUV. The trajectory is therefore planned in 3D. The follower creates a helix-like trajectory around the leader, as can be seen in figure 5.



(a) (b)

(c) (d)

Fig. 4: **(a)**: ASV with a constant speed of 5m/s, with possible turning radii of 25 and 50m following an AUV. The follower gets updated at the start of each new lawnmower leg by the leader with the leader's position, direction and speed. The follower plans a path to minimize the average distance between the leader and follower over time.
(b): Distance and angle between ASV and AUV for the scenario in (a). At different speeds the relationship between the two vehicles change continuously over time, giving a good base for trilateration.
(c): Part of a simulation where the ASV and the AUV have a constant speed of 1m/s. The AUV has a possible turning radii of 25 and 50m. At the first corner the ASV falls behind when the AUV has turned and indicated a new direction. At the next corner the ASV will cut a bit to catch up.
(d): Distance and angle between ASV and AUV for the scenario in (c). Relationship is steady during legs and changes when turning. The AUV is performing the same lawnmower survey in both scenarios.

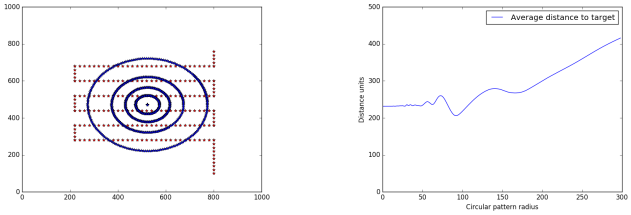


(a) (b)

Fig. 5: One AUV trying to decrease the average distance between itself and another AUV that is moving in a line. The follower creates a helix-like pattern around the target.

Algorithm 1 Kinodynamic Path Follower

```
1:  procedure KPF($state, pattern, environment, targetStart, speedVector, time$)
2:      $openList, ClosedList = \emptyset, \emptyset$
3:      $openList.insert(state)$
4:      while $openList$ do
5:          $\_State = openList.pop()$
6:          if $\_State.time => time$ then /* Found final condition */
7:              break
8:          for $branch$ in $pattern$ do
9:              $accepted = True$
10:             for $wp$ in $branch$ do
11:                 $intermediate = getGlobalState(wp, \_State)$
12:                 if not $acceptState(intermediate, environment)$ then
13:                     $accepted = False$
14:             if $accepted$ then
15:                 $newState = state(intermediate)$ /* Last state of intermediate States for the leg */
16:                 $newState.parent = \_State$
17:                 $newState.f = f(newState)$ /* see equation 5 */
18:                 $openList.insert(newState)$
19:         $closedList.insert(\_State)$
20:         $sortList(openList)$ /* based on f value at line 17 */
21:     return $\_State$
```



(a)

(b)

Fig. 6: ASV performing a circular pattern with a speed of 5m/s around center point of the survey. The AUV runs at a constant speed of 2m/s and is performing a lawnmower pattern as seen in figure 4a.
(b) average distance between vehicles with different radii of the ASV.

V. CONCLUSION AND FUTURE WORK

This paper describes a new approach to path planning for the leader-follower paradigm in the maritime domain. The approach takes into account kinodynamic constraints and can be used for AUVs and ASVs. It uses a highly modified version of HA*, where the function minimizes the average distance between the vehicles along their respective paths across a specific time window. Simulations in 2-Dimensional (2D) and 3D environments have shown promising results. Vehicles under different kinematic constraints are able to track and follow other vehicles whilst avoiding collisions with known obstacles in the environment.

We are now planning to validate the algorithms on real platforms in open water trials. The next step in terms of algorithm development is to extend it for multiple vehicles, both on the surface and subsurface.

VI. ACKNOWLEDGMENTS

This work was supported by the EPSRC funded ORCA RAI-HUB (EP/R026173/1) and USMART projects (EP/P017975/1) as well as the EU H2020 Strongmar project (H2020-TWINN-2015, 692427).

REFERENCES

[1] A. Vasilijević, D. Nad, F. Mandi, N. Miškovi, and Z. Vukić, “Coordinated navigation of surface and underwater marine robotic vehicles for ocean sampling and environmental monitoring,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 3, pp. 1174–1184, 2017.

[2] J. S. Willners, P. Patron, and Y. R. Petillot, “Moving Baseline Localization for Multi-Vehicle Maritime Operations,” *OCEANS 2017 - Aberdeen*.

[3] F. Mandi, “Underwater navigation and localization using single range measurements,”

[4] R. M. Eustice, H. Singh, and L. L. Whitcomb, “Synchronous-clock, one-way-travel-time acoustic navigation for underwater vehicles,” *Journal of Field Robotics*, vol. 28, no. 1, pp. 121–136, 2011.

[5] J. Melo and A. Matos, “Guidance and control of an ASV in AUV tracking operations,” *Oceans 2008*, 2008.

[6] A. Matos and N. Cruz, “Coordinated operation of autonomous underwater and surface vehicles,” *Oceans Conference Record (IEEE)*, pp. 2–7, 2007.

[7] M. Bibuli, O. Parodi, L. Lapierre, G. Bruzzone, and M. Caccia, *Vehicle-following guidance for unmanned marine vehicles*, vol. 42. IFAC, 2009.

[8] M. Kumru, K. Leblebiciolu, I. K. Erünsal, and K. Ahiska, “A survey on tactical control algorithms for path tracking unmanned surface vehicles,” *2016 14th International Conference on Control, Automation, Robotics and Vision, ICARCV 2016*, vol. 2016, no. November, pp. 13–15, 2016.

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

[10] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *Systems Science and Cybernetics*, no. 2, pp. 100–107, 1968.

[11] N. D. Richards and M. Sharma, “A Hybrid A*/Automaton Approach to on-Line Path Planning with Obstacle Avoidance,” *AIAA*, pp. 141–157, 2004.

[12] J. Pan, S. Chitta, and D. Manocha, “FCL: A general purpose library for collision and proximity queries,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2012.

[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” *Autonomous Robots*, 2013.

[14] A. Bahr, J. J. Leonard, and M. F. Fallon, “Cooperative localization for autonomous underwater vehicles,” *International Journal of Robotics Research*, vol. 28, no. 6, pp. 714–728, 2009.