Path following algorithm for minimally specified lawn-mower type AUV missions

José Mare
ATSA
21 Huntingdale Drive
Thornton, NSW 2322, Australia
Email: jose.mare@atsa.com.au

***Abstract*— This paper describes a simple online target point generator algorithm that allows for minimal specification of Autonomous Underwater Vehicle (AUV) missions. The mission pattern is specified through the spatial coordinates of its corners only, requiring minimal mission information storage in order to execute a mission. To follow the mission path the algorithm generates intermediate target points along the desired trajectory based on the current estimated position of the AUV. The performance of the algorithm when implemented and tested in a real AUV is reported.**

I. Introduction

AUV missions often involve the tracking of straight lines. For example, rectangular patterns and lawn-mower patterns, which are combinations of straight line paths, are common in sea floor surveying applications. For missions described only by corner coordinates, the navigation algorithm needs to generate way-points in order not to miss the segment it is tracking if the AUV drifts away from it. To illustrate this, consider Figure 1: Figure 1.a shows a mission specified by the points A, B, C and D. In Figure 1.b the AUV has instantly drifted away from the segment $\overline{AB}$ and, while aiming at point B, misses the remainder of the segment. It is important to stress that for an AUV carrying a payload such as a camera or a sonar, missing the trajectory path can spoil the mission. Figure 1.c introduces fixed way-points which help the vehicle return to the segment $\overline{AB}$. In Figure 1.d the vehicle has drifted away from the trajectory and one of the way-points has to be skipped so the vehicle keeps travelling forward. In some situations way-points also have to be skipped to reduce the oscillation around the desired path due to the AUV dynamics.

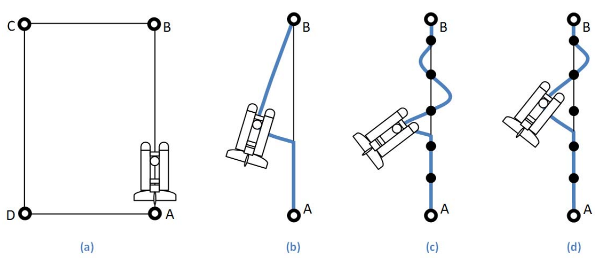

Fig. 1. Fixed Way-point Scenarios

The use of fixed way-points equally spaced along the desired pattern constitutes a standard approach to path following (see for example [3]. For a detailed discussion on way-point generation, see [1], Chapter 5.) When the mission is executed, the AUV navigation algorithm sequentially aims at each way-point. Despite the simplicity of generating fixed equally spaced way-points, one of the disadvantages of its implementation is that the navigation algorithm needs to determine which way-points are skipped as the mission evolves. The algorithm presented in this paper avoids this decision making process by generating, in mission-execution time, target points that are not fixed, but whose location rather depends on the estimated position of the AUV. These target points are generated in a way that ensures that the vehicle continually travels forward. The algorithm can also be tuned to help reduce the oscillation of the vehicle around the desired mission pattern.

The reminder of the paper is organised as follows: Section II presents the algorithm that generates the target points for path following and discusses design choices. Section III describes the implementation of the algorithm in a real AUV that is currently being developed by ATSA. The performance of the algorithm executing lawn-mower type missions is also reported in that section. Finally, conclusions are drawn in Section IV.

II. The target point generator

In this section we introduce the target point algorithm. An illustrative example and a discussion on design choices are also the subject of this section.

A. *The algorithm*

Let us assume that at every sampling time the AUV has an estimate of its current position $\hat{x} \in \mathcal{R}^n$ (with $n = 2$ for 2-dimensional mission paths, or $n = 3$ for 3-dimensional mission paths). The estimate $\hat{x}$ is obtained using measurements from the on board instruments (see [2] for traditional model-based estimation techniques. The modelling of AUVs is treated in, e.g., [1], [4] and [5].) Given a desired trajectory line $\overleftrightarrow{AB}$ defined by two points $A, B \in \mathcal{R}^n$, our goal is to make the AUV travel towards point B along the path $\overleftrightarrow{AB}$. The path following algorithm proposed here finds, for each position estimate $\hat{x}$, a target point on the trajectory $\overleftrightarrow{AB}$. By sequentially aiming at these target points, for suitable choices of the design parameters, the AUV converges to the desired trajectory $\overleftrightarrow{AB}$.

The target point generator can be divided into 2 steps:

978-1-4244-5222-4/10/$26.00 ©2010 IEEE

1) Find $P(\hat{x})$, the orthogonal projection of the current AUV estimated position $\hat{x}$ onto the desired trajectory $\overleftrightarrow{AB}$.
2) Obtain a target point $T(\hat{x})$, corresponding to the AUV position $\hat{x}$, as the point on the desired trajectory such that satisfies the following condition:

$$T(\hat{x}) = \arg\min_{p\in\Omega} \|B - p\|, \qquad (1)$$

$$\Omega = \{p \in \overleftrightarrow{AB} \text{ s.t } \|P(\hat{x}) - p\| = d\}. \qquad (2)$$

where $d > 0$ is a design parameter.

This condition implies that the target point $T(\hat{x})$ is the point along the mission path $\overleftrightarrow{AB}$, at a distance d from $P(\hat{x})$, that is the closest to point B. Notice that the solution of the algorithm always exists: the set Ω in equation (2) always contains 2 elements, and equation (1) yields a unique solution unless the projection $P(\hat{x})$ coincides with B (in which case it yields two solutions. We will discuss this case in Section III.)

B. Illustrative Example

Figure 2 shows a 2-dimensional example of the implementation of the algorithm. The example is a section of an actual mission carried out by a real AUV. The algorithm runs every 400 ms, and the AUV travels at a controlled speed of 0.5 m/s and a controlled altitude of 1.5 m. The choice of the design parameter is $d = 1$ m. The desired mission path is shown by the vertical segment in the figure. The initial position of the AUV is $\hat{x}$. Finding the projection of $\hat{x}$ onto the desired path, $P(\hat{x})$, a first target point $T(\hat{x})$ is obtained as illustrated in the figure. The AUV will start traveling towards $T(\hat{x})$, reaching a new position. The actual AUV trajectory is indicated with solid circles. The algorithm will continue computing target points (indicated by the $\times$ markers along the desired path) as the AUV moves aiming at them. It can be seen from Figure 2 that the vehicle converges to the desired path.

C. The Design Parameter d

When the AUV obtains an estimate of its position, it needs to determine what direction to travel in order to return to the required mission path. This direction becomes the setpoint (SP) of the Heading controller implemented in the AUV.

The target point algorithm in this paper can be thought of as a modulator of the Heading SP. When the vehicle drifts away from the desired path, the algorithm provides a Heading SP that depends on the choice of the design parameter d in equation (2). A very small value of d will generate rapidly changing SPs with large error (error = actual Heading - SP). This can produce undesirable oscillations around the mission path. On the other hand, a very large value of d will render more steady SPs with generally smaller error, but can result in a very slow return to the mission path. For a given choice of d, the SPs the algorithm generates become more steady as the AUV returns to the desired path.

The effect the choice of d has on how the AUV returns to the trajectory depends on two main factors:

- The sampling time (how often the algorithm generates target points)

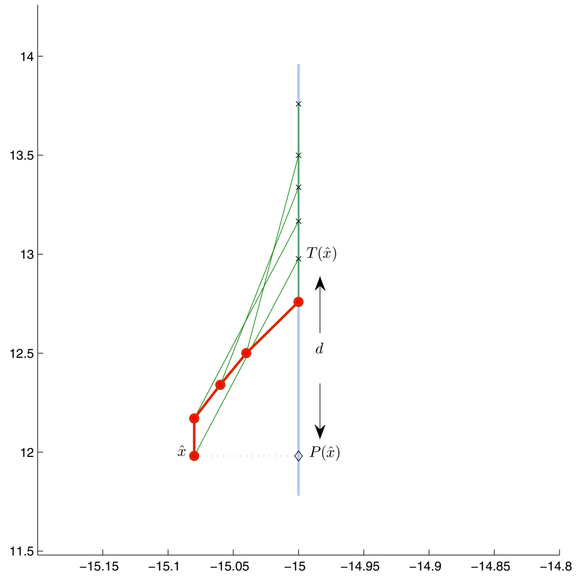

Fig. 2. Target Point Generator

- The speed the vehicle travels.

A possible modification of the algorithm could be the implementation of the design parameter d as a function of the AUV travel speed (or the speed SP), i.e., $d(v)$.

III. Implementation of the Target Point Algorithm in a Real AUV

The path following algorithm presented in the previous section has been implemented and tested in a real AUV that is currently being developed by ATSA in Thornton, NSW, Australia. In the following subsections we will describe the main characteristics of ATSA's AUV, some of the design decisions made in order implement the algorithm, and finally present and discuss the results obtained executing lawn-mower type missions in recent trials.

A. The SeaVision AUV

ATSA's SeaVision AUV, shown in Figure 3, is a twin-hull design. Approximate dimensions of the current prototype are summarised in Table I.

TABLE I
AUV Approximate Dimensions

Length	1.10 m
Width	0.70 m
Height	0.30 m
Mass	25 Kg

The twin hulls contain all the electronics of the vehicle and the connectors to interface the peripherals. Two thrusters

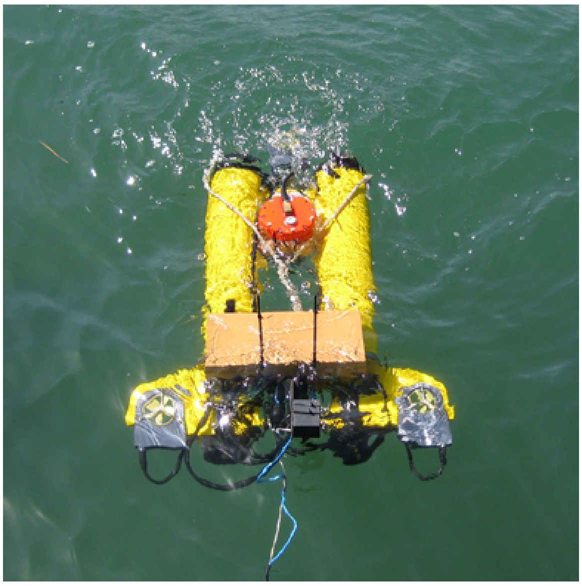

Fig. 3. ATSA's SeaVision AUV

are used for the horizontal motion (surge and yaw). Acting in opposition, these thrusters make the vehicle turn. Three thrusters are used to control the vertical motion (heave, roll and pitch). When the two vertical thrusters at the back act in opposition, the vehicle rolls. When the two vertical thrusters at the back work together, but in opposition to the vertical thruster at the front, the vehicle pitches.

The twin-hull design is stable in roll and pitch, requiring only minor control action in those degrees of freedom. A feature that arises from this is the vehicle's inherent capability to hover. This capability is useful when executing lawn-mower type missions, as the vehicle can actually stop and turn at the end of each run.

The SeaVision AUV is intended as a platform to carry a payload between the twin hulls. Payloads such as underwater cameras, sonars and other sensors impose limitations on the maximum speed they can move. Therefore, the SeaVision AUV is designed to operate at low speeds. Typical missions are executed with a speed setpoint of 1 knot. Traveling at low speeds allows for the decoupling of the vehicle dynamics, which simplifies the control of the AUV. ATSA's prototype features control of altitude/depth, speed, pitch, roll and yaw.

The main navigation instrument in the AUV is a Doppler Velocity Log (DVL). The DVL has four beams that ping the sea floor to determine the speed the AUV is traveling at. It also measures altitude above the sea floor and has an integrated magnetic compass, which is tilt compensated, used to determine yaw. The DVL also provides measurements of pitch and roll, among other variables.

The estimation of the current position $\hat{x}$ of the AUV is performed via dead-reckoning from a known launch location (regarded as the origin of coordinates). As the vehicle moves, its main computer filters and integrates the velocity measurements from the DVL so as to determine its new position. These estimates of the position are used in the target point algorithm.

B. Implementation Issues

The mission files implemented in the SeaVision AUV prototype are organised as a list of goals that are executed in order. The goals only contain the travel speed setpoint, the altitude or depth setpoints, and the corner coordinates of the path the AUV is intended to follow. The following is an example of such mission files:

```
#This mission is a square
START:45,,,END #start here in 45 seconds!
GOAL0:0.5,1.5,,10,0,END
GOAL1:0.5,1.5,,10,-10,END
GOAL2:0.5,1.5,,0,-10,END
GOAL3:0.5,1.5,,0,0,END #End of mission.
```

Since the vertical and horizontal controllers are decoupled, the mission pattern can be regarded as a 2-dimensional trajectory that the vehicle tracks while simultaneously controlling altitude and speed.

The tracking of the 2-dimensional trajectory is executed as follows: First, the AUV calculates the line $\overleftrightarrow{AB}$ between the previous goal (or launch point, initially) and the next goal listed in the mission file. Then, before the vehicle starts traveling forward, the vehicle turns until it is aiming at the next target point within a certain tolerance. In order to have a more steady heading SP during this heading correction stage, the design parameter d in equation (2) is temporarily set at a large value (e.g., $d = 10$ m; see discussion in Section II-C). This choice of a large d reduces the time it takes the vehicle to reach the heading SP and start traveling forward. Once the heading SP has been reached and the vehicle starts tracking $\overleftrightarrow{AB}$, d is set to a smaller value (e.g., $d = 1$ m). This sequence is repeated at every corner.

When the AUV is approaching a goal (point B in the algorithm, as shown in Figure 4) as the projection $P(\hat{x})$ reaches a certain distance from B, but before it actually reaches point B (dashed cirlce in Figure 4), the vehicle looks for the next goal in the mission file and starts aiming at the new path $\overleftrightarrow{AB}$. In this way, the case discussed in Section II-A, where the algorithm could yield two solutions, is avoided.

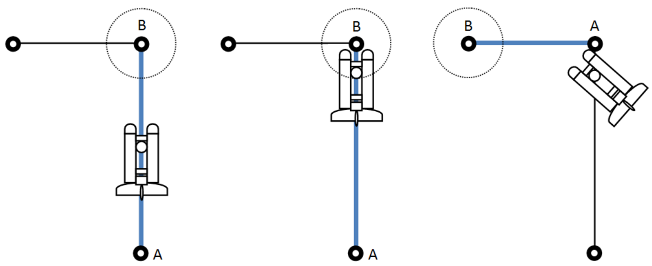

Fig. 4. Corner Handling

C. Results

The path following algorithm has been implemented in the SeaVision AUV as discussed above. To illustrate its performance, we present a lawn-mower type mission of a total length of 345 m. Over the whole mission, the altitude setpoint is 1.5m and the speed setpoint is 0.5 m/s. An estimate of the AUV position is obtained every 200 ms, and the target point algorithm updates the target point (or, equivalently, the heading setpoint) every 400 ms. Below is the mission file corresponding to the lawn-mower pattern:

```
#Lawnmower pattern 40x30m
START:45,,,END #start here in 45 seconds!
GOAL0:0.5,1.5,,0,-15,END
GOAL1:0.5,1.5,,40,-15,END
GOAL2:0.5,1.5,,40,-25,END
GOAL3:0.5,1.5,,0,-25,END
GOAL4:0.5,1.5,,0,-35,END
GOAL5:0.5,1.5,,40,-35,END
GOAL6:0.5,1.5,,40,-5,END
GOAL7:0.5,1.5,,30,-5,END
GOAL8:0.5,1.5,,30,-35,END
GOAL9:0.5,1.5,,20,-35,END
GOAL10:0.5,1.5,,20,-5,END
GOAL11:0.5,1.5,,10,-5,END
GOAL12:0.5,1.5,,10,-35,END
GOAL13:0.5,1.5,,0,-35,END
GOAL14:0.5,1.5,,0,-5,END #End of mission.
```

This mission was executed in 15 minutes and 30 seconds. The SeaVision AUV logs some navigational data that allows for replaying the missions in a software simulator and analysing the behaviour of the AUV. Figure 5 shows the desired mission path and the actual trajectory the AUV followed. The figure also indicates (in thicker trace) the sections of the trajectory where the DVL could not obtain valid readings. In such sections the estimated position of the AUV is more inaccurate. Notice in particular the NE corner, where the loss of bottom lock happened very regularly.

It can be seen from Figure 5 that the algorithm effectively keeps the AUV on track. The maximum deviation from the intended trajectory happens around the corners, and in the worst case is approximately 1.2 m. Along the straight sections, the vehicle remains within a few centimetres of the desired path, without exhibiting excessive oscillation around it.

Figure 6 shows the performance of the Speed and Altitude controllers that run independently from the target point generator algorithm. The frequent stops on each corner can be seen in the speed plot at the top of Figure 6, where the speed is reduced. Along the straight sections of the mission the speed is kept around the setpoint specified in the mission file. The noisy section between samples 1000 and 1500 corresponds to the NE corner in Figure 5, where the DVL could not obtain bottom lock.

The bottom plot in Figure 6 shows the Altitude control. The vehicle is launched in shallow water (approximately 2m deep). The Altitude controller keeps the AUV's altitude within 30 cm

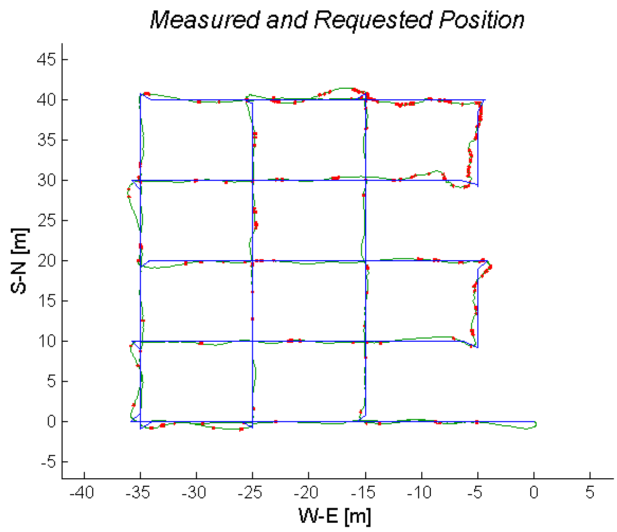

Fig. 5. Tracking of a Lawn-Mower Pattern

of the altitude setpoint specified in the mission file. Again, between samples 1000 and 1500 corresponding to the NE corner in Figure 5, the altitude measurement was inaccurate due to the loss of bottom lock.

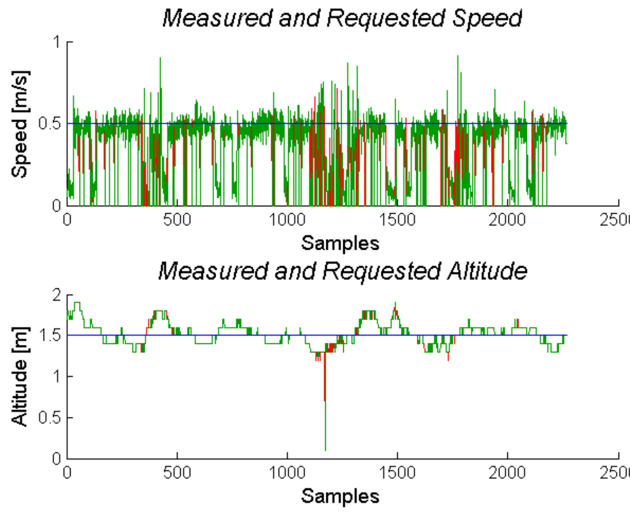

Fig. 6. Speed and Altitude Control

IV. Conclusion

This paper has presented a simple online target point generator algorithm that allows for minimal specification of Autonomous Underwater Vehicle (AUV) missions, while ensuring that the AUV travels along the desired mission path. The design choices and the algorithm implementation issues have been discussed. The algorithm has been implemented and tested in a real AUV, and its performance executing lawn-mower type missions has been reported. Some directions from future work have been discussed, such as the selection of a design parameter d as a function of the AUV speed.

References

[1] T. Fossen, *Marine Control Systems. Guidance, Navigation and Control of Ships, Rigs, and Underwater Vehicles*, 1st ed. Trondheim, Norway: Marine Cybernetics, 2002.

[2] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems, Transaction of the ASME–Journal of Basic Engineering, 1960, pp. 35–45.

[3] T. Holzhüter, "LQG approach for the high-precision track control of ships", IEE Proceedings on Control Theory and Applications 144(2), 1997, pp. 121–127.

[4] J. Yuh, "Modeling and Control of Underwater Robotic Vehicles", IEEE Transactions on Systems, Man, and Cybernetics 20(6), 1990, pp. 1475–1483.

[5] F. Song, P. Edgar An and A. Folleco, "Modelling and Simulation of Autonomous Underwater Vehicles: Design and Implementation", IEEE Journal of Oceanic Engineering 28(2), 2003, pp. 283– 296.