Abstract - This paper presents recent work in the areas of simulation, mission planning, and mission execution for an unmanned underwater vehicle (UUV). The UUV we consider is the Manta Test Vehicle (MTV), operated by the Naval Undersea Warfare Center (NUWC) in Newport, RI. A 6-Dof Simulink model of the MTV vehicle dynamics augmented with an autopilot is used to test the algorithms.

The on-board mission planner generates reference trajectories for the vehicle to follow, taking into consideration bathymetry data and moving obstacles that are within the forward-looking sonar range. A trajectory consists of a sequence of waypoints and associated headings from the current vehicle location and orientation to the goal. Trajectory generation takes into consideration the dynamic capabilities of the MTV. The D* algorithm - an extension to the Dijkstra shortest-path algorithm which allows efficient re-planning when arc-costs change - is used to generate and maintain a safe trajectory. Trajectory re-planning is triggered when the sonar detects an obstacle in the trajectory currently being followed.

A Model Predictive Control (MPC) algorithm is inserted between the D* algorithm and the vehicle inner loop autopilot. The MPC algorithm issues the reference commands to the autopilot to allow the vehicle to follow the planned trajectory. The cost function within the MPC algorithm can be changed depending on the guidance task. The MPC algorithm uses a full nonlinear model of the MTV vehicle to project ahead the output trajectory and employs orthogonal Laguerre polynomials to create basis functions that are used in the synthesis of reference commands to the autopilot. The MPC controller also provides a second layer of obstacle avoidance capability and keeps the vehicle on-track in the presence of a current.

I. INTRODUCTION

Since 1996 the US Navy’s Naval Undersea Warfare Center (NUWC) has been developing a concept for underwater warfare in the new century. Playing a major role in this futuristic concept are mission-reconfigurable unmanned underwater vehicles (MRUUVs) that will be capable of carrying out surveillance, tactical oceanography, mine warfare, and anti-submarine warfare missions. The Manta Test Vehicle (MTV) is the first prototype.

The missions require UUVs to operate within the area of simulation, mission planning, and mission execution for an unmanned underwater vehicle (UUV). The UUV we consider is the Manta Test Vehicle (MTV), operated by the Naval Undersea Warfare Center (NUWC) in Newport, RI. A 6-Dof Simulink model of the MTV vehicle dynamics augmented with an autopilot is used to test the algorithms.

The on-board mission planner generates reference trajectories for the vehicle to follow, taking into consideration bathymetry data and moving obstacles that are within the forward-looking sonar range. A trajectory consists of a sequence of waypoints and associated headings from the current vehicle location and orientation to the goal. Trajectory generation takes into consideration the dynamic capabilities of the MTV. The D* algorithm - an extension to the Dijkstra shortest-path algorithm which allows efficient re-planning when arc-costs change - is used to generate and maintain a safe trajectory. Trajectory re-planning is triggered when the sonar detects an obstacle in the trajectory currently being followed.

A Model Predictive Control (MPC) algorithm is inserted between the D* algorithm and the vehicle inner loop autopilot. The MPC algorithm issues the reference commands to the autopilot to allow the vehicle to follow the planned trajectory. The cost function within the MPC algorithm can be changed depending on the guidance task. The MPC algorithm uses a full nonlinear model of the MTV vehicle to project ahead the output trajectory and employs orthogonal Laguerre polynomials to create basis functions that are used in the synthesis of reference commands to the autopilot. The MPC controller also provides a second layer of obstacle avoidance capability and keeps the vehicle on-track in the presence of a current.
**Report Documentation Page**

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>01 SEP 2003</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. REPORT TYPE</td>
<td>N/A</td>
</tr>
<tr>
<td>3. DATES COVERED</td>
<td>-</td>
</tr>
<tr>
<td>4. TITLE AND SUBTITLE</td>
<td>UUV On-Board Path Planning in a Dynamic Environment for the Manta Test Vehicle</td>
</tr>
<tr>
<td>5a. CONTRACT NUMBER</td>
<td></td>
</tr>
<tr>
<td>5b. GRANT NUMBER</td>
<td></td>
</tr>
<tr>
<td>5c. PROGRAM ELEMENT NUMBER</td>
<td></td>
</tr>
<tr>
<td>5d. PROJECT NUMBER</td>
<td></td>
</tr>
<tr>
<td>5e. TASK NUMBER</td>
<td></td>
</tr>
<tr>
<td>5f. WORK UNIT NUMBER</td>
<td></td>
</tr>
<tr>
<td>6. AUTHOR(S)</td>
<td>Draper Laboratory Aerospace Control Group 555 Technology Square Cambridge, MA, 02139, US</td>
</tr>
<tr>
<td>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</td>
<td>Draper Laboratory Aerospace Control Group 555 Technology Square Cambridge, MA, 02139, US</td>
</tr>
<tr>
<td>8. PERFORMING ORGANIZATION REPORT NUMBER</td>
<td></td>
</tr>
<tr>
<td>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</td>
<td></td>
</tr>
<tr>
<td>10. SPONSOR/MONITOR’S ACRONYM(S)</td>
<td></td>
</tr>
<tr>
<td>11. SPONSOR/MONITOR’S REPORT NUMBER(S)</td>
<td></td>
</tr>
<tr>
<td>12. DISTRIBUTION/AVAILABILITY STATEMENT</td>
<td>Approved for public release, distribution unlimited</td>
</tr>
<tr>
<td>14. ABSTRACT</td>
<td></td>
</tr>
<tr>
<td>15. SUBJECT TERMS</td>
<td></td>
</tr>
<tr>
<td>16. SECURITY CLASSIFICATION OF:</td>
<td></td>
</tr>
<tr>
<td>a. REPORT</td>
<td>unclassified</td>
</tr>
<tr>
<td>b. ABSTRACT</td>
<td>unclassified</td>
</tr>
<tr>
<td>c. THIS PAGE</td>
<td>unclassified</td>
</tr>
<tr>
<td>17. LIMITATION OF ABSTRACT</td>
<td>UU</td>
</tr>
<tr>
<td>18. NUMBER OF PAGES</td>
<td>8</td>
</tr>
<tr>
<td>19a. NAME OF RESPONSIBLE PERSON</td>
<td></td>
</tr>
</tbody>
</table>

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18
used to describe the vehicle’s dynamics: \( \mathbf{v} \) is a vector of linear and angular rates in the body-fixed frame; \((x, y, z)\) is the position of the vehicle’s origin relative to the earth-fixed frame; \(\phi, \theta,\) and \(\psi\) are the roll, pitch, and yaw angles (i.e., the Euler angles).

![Fig. 1 NUWC's Manta Test Vehicle (MTV) and coordinate frame definitions.](image)

The control input vector is 

\[
\mathbf{u} = [u \ \delta_a \ \delta_e \ \delta_r]^T,
\]

where \(u > 0\) is the speed of the propeller, and \(\delta_a, \delta_e, \delta_r\) are the angular positions of the aileron, elevator, and rudder control surfaces. With these definitions, the state equations are

\[
\dot{\mathbf{v}} = \mathbf{M}^{-1} [\mathbf{h}(\mathbf{v}) - \mathbf{C}_{RB}(\mathbf{v}) \mathbf{v} - \mathbf{g}(\eta) + \mathbf{\tau}(\mathbf{v}, \mathbf{u}) + \tau_E],
\]

\[
\dot{\eta} = \mathbf{J}(\eta) \mathbf{v},
\]

where \(\mathbf{M}\) is the inertia matrix (including the inertia of the water surrounding the vehicle), \(\mathbf{h}(\mathbf{v})\) is a vector of hydrodynamic forces and moments due to body rates only, \(\mathbf{C}_{RB}(\mathbf{v})\) is the rigid-body Coriolis and centripetal matrix, \(\mathbf{g}(\eta)\) is the restoring forces and moments vector (i.e., the effects of gravity and buoyancy), \(\mathbf{\tau}(\mathbf{v}, \mathbf{u})\) is the vector of hydrodynamic forces and moments due to control surface deflections and propeller speed, and \(\tau_E\) represents environmental forces and moments (i.e., due to ocean currents, waves, etc.). \(\mathbf{J}(\eta)\) is the body-to-earth kinematic transformation.

Each element of the vectors \(\mathbf{h}(\mathbf{v})\) and \(\mathbf{\tau}(\mathbf{v}, \mathbf{u})\) is expressed as a linear combination of functions that are appropriately chosen to best describe the hydrodynamics of the vehicle. The coefficients of each linear combination and the hydrodynamic derivatives in \(\mathbf{M}\) (i.e., the added-mass coefficients) are jointly called the hydrodynamic coefficients, and are determined experimentally and via numerical methods. Typically, several sets of hydrodynamic coefficients are produced, each corresponding to a certain operating condition of the vehicle. In the case of MTV, three sets of coefficients have been obtained: 2.5-knots, 5-knots, and 10-knots coefficients.

The mapping between the MTV’s actual vehicle control surfaces, \(\delta = [\delta_{a_{top}} \ \delta_{e_{top}} \ \delta_{e_{bot}} \ \delta_{r_{top}}]^T\), and the “virtual” ones, \(\delta_v = [\delta_a \ \delta_e \ \delta_r]^T\), used in the 6-Dof equations of motion, is

\[
\delta = \begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix} \delta_v. \tag{3}
\]

Each actual control surface is driven through various joints and linkages by a high-performance permanent-magnet DC motor. Control surface displacement is achieved by controlling the armature voltage, \(u_a\), applied to the motor windings.

The MTV’s propeller is driven by a high-performance permanent-magnet synchronous motor. More details on the modeling and control of this motor can be found in [3]. The nonlinear torque load due to the hydrodynamics of the propeller is

\[
Q(n,V_e) = Q_m n^2 + Q_{nv} n V_e, \tag{4}
\]

where \(Q_m > 0\) and \(Q_{nv} < 0\) are design parameters that depend on propeller geometry and other hydrodynamic variables; \(V_e\) is the speed of the water going into the propeller, which is typically a fraction of the vehicle’s forward speed, \(u\). Similar to the control surface dynamics, electromechanical constraints impose limits on the ranges of \(u_a\) and \(n\), and these constraints are included in the vehicle model.

The MTV sensor suite is composed of many subsystems, including a strapdown ring-laser gyro Inertial Navigation System (INS), a phased-array Doppler Velocity Sonar (DVS), a depth sensor, and a differential GPS receiver. Sensor fusion based on Kalman filtering is performed to deliver measurements of every state variable of interest (i.e., \(v, \eta, \delta\) and \(n\)). All the available information on sensor dynamics, filtering, quantization values, dynamic ranges, noise power levels, etc., has been incorporated in the nonlinear model.

### III. PATH PLANNING AND CONTROL

A conventional vehicle control system has two independently designed subsystems, a guidance system and a flight control system. Traditionally the guidance is responsible for generating suitable guidance commands for the inner loop autopilot. The autopilot, also called the flight control system,
keeps the vehicle on the trajectory commanded by
the guidance algorithm.

In the current design there is not such a clear
distinction between the two functions. In general
the D* and the B-Spline algorithms can be viewed
as the guidance function and the MPC and the
inner loop autopilot as the control function. Fig. 2
shows the Simulink implementation of the MTV
control system described in this paper.

The path planning and control scheme
presented in this paper is divided in 3 stages.

In the first stage a minimum path is computed
by a D* algorithm from the current UUV position to
the target position based on bathymetry data and
the current location of the obstacles. The path
generated by the D* algorithm cannot be directly
tracked by the control function because of the
sharp corners connecting consecutive segments.

The second stage uses a B-Spline algorithm to
generate a trajectory that the UUV can fly over the
next \( N \) waypoints. The optimum B-Spline path is
generated incrementally as the vehicle proceeds
along the trajectory.

The third stage implements a Model Predictive
Control algorithm to generate the steering
commands that keep the UUV on the desired
trajectory.

The three stages are described in detail in the
next sections.

A. D* Planning Algorithm

The responsibility of the mission planner is to
maintain a list of future waypoints for the MPC
algorithm to follow. These waypoints must define
a path which the test vehicle is capable of tracking
and which allows the vehicle to travel at its
commanded depth while avoiding collisions with
moving obstacles.

Our path planning algorithm operates on a 5 km
square map of bathymetry data (divided into 50m
cells) for a section of the Narragansett Bay, RI.
For the purposes of path planning, the vehicle
state is defined by its cell and one of eight possible
headings. The legal state transitions are shown in
Fig. 3. From any given state there is a maximum of	hree neighboring states.

Obstacle Avoidance

The vehicle’s sensors can determine the
traversability of any cell within a specified distance
from the current vehicle position. If an obstacle is
detected along the vehicle’s planned path, the map
is updated and the re-planning function is called.
In Fig. 4 an obstacle is shown crossing the
vehicle’s planned path. However, given the limited
vehicle sensor range this obstacle goes
undetected and no replanning is performed.
In Fig. 5, the vehicle has moved forward, bringing the obstacle within sensor range. The obstacle is detected and the map is updated to reflect the cells that are now invalid (X=37 and Y=70,71,72). The planner strategy is to conduct a simple re-plan the first time an obstacle is detected. The vehicle then follows the new path. In Fig. 5 we see the vehicle following the new path passing in front of the obstacle, the dotted line represents the old path.

However, if the new path becomes blocked again by the same obstacle (in case of a dynamic obstacle), an alternate path is planned. Different strategies can be implemented to evade a blocking dynamic obstacle. Fig. 6. shows a simple strategy that brings the vehicle behind the obstacle. Once the point behind the vehicle is reached a new path to the target is calculated.

One of the major concerns in using the D* algorithm in a real time application is computation speed. Tab. 1 shows the processing time required for these scenarios, running on a pentium IV (2.2GHz). The D* algorithm is currently executed at 0.1 Hz, this gives a margin to increase the planning search space or the rate at which the algorithm is executed.

**B. B-Spline Reference Trajectory**

The trajectory calculated by the D* algorithm is a set of waypoints connected by path segments. Two adjacent segments can lie on the same line or at angle of ±45 degrees. Fig. 7 shows an hypothetical trajectory with 12 waypoints.

The B-Spline is a technique used in computer graphics to generate smooth curves. An introduction to spline curves can be found in reference [9].

The B-Spline algorithm is applied to 5 waypoints in the D* trajectory. The first control point is the waypoint just behind the current location of the vehicle. In Fig. 7 large filled circles indicate the spline control points, the X indicate past waypoints, and empty circles future waypoints. Fig. 7 also shows how the degree $T$ of the spline affects the shape of the curve. For $T = 1$ the spline curves are just connecting the waypoints with straight segments, increasing the degree we get smoother curves between the waypoints. The distance between the waypoints and the degree of the spline are selected based on the dynamics of the vehicle.
Fig. 7: B-Spline Curves

Fig. 8 shows the response of the MTV vehicle to a step command in the y (North) direction. The vehicle has difficulties tracking low degree B-Spline (order 1 or 2), while it is able to well track B-Splines of order 3 and 4. Higher order splines generate smoother trajectories and can be used in areas where it is not required to remain close to the segments connecting the waypoints. The order of the spline can be changed by the autopilot during the mission.

Once the B-Spline curve over the next 5 waypoints is generated it is passed to the MPC. The MPC is then responsible to track the trajectory as close as possible. In the MTV implementation the B-Spline algorithm is running at 2Hz, the same rate as the MPC and it is incorporated in the block called MPC in Fig. 2.

MPC is based on the following points:

- At each instant $t$, the future outputs $\tilde{y}(t + j)$ are predicted over a determined finite horizon $T_p$. An internal model of the plant is used by MPC to predict the future output of the plant. In this application the internal model is nonlinear.

$$\tilde{x}(t) = f(x(t), u(t))$$
$$\tilde{y}(t) = g(x(t), u(t))$$

- The future control signals are calculated by optimizing a performance index to keep the process as close as possible to a reference trajectory $y_r(t)$. In the current application the index is a weighted quadratic function of the squared distance between the predicted output trajectory and the reference trajectory ($\tilde{d}(\tau)$).

The optimization problem is

$$\min_{\tilde{u}^*} \int J(y(t), \tilde{u}(t); T_p)$$
$$J(y(t), \tilde{u}(t); T_p) = \int_{0}^{T_p} G(\tilde{y}(\tau), \tilde{u}(\tau)) d\tau$$

subject to equation (5) and the following constraints:

$$\tilde{u}(\tau) \in U, \quad \forall \tau \in [t, t + T_p]$$
$$\tilde{y}(\tau) \in Y, \quad \forall \tau \in [t, t + T_p]$$

C. MPC Based Autopilot

The MPC is responsible for tracking the B-Spline reference trajectory by generating yaw rate commands for the inner loop autopilot. The reference trajectory is defined in terms of x (East) and y (North) positions on the map.

The MPC methodology is based on the on line solution of a finite horizon open loop control problem subject to plant dynamics and constraints on states, outputs, and inputs. In our application the plant is formed by the close loop autopilot-vehicle system.

Fig. 9 shows the basic principle of model predictive control.
where $T_p$ is the prediction horizon. $Q$ in equation 6 is the weighting matrix. $Q$ is a diagonal matrix that allows different weightings of present and future errors.

In equation 6 the squared distance between the reference trajectory and the predicted trajectory is defined as

$$\overline{d}(\tau) = (\xi(\tau) - \xi_n(\tau))^2 + (\eta(\tau) - \eta_n(\tau))^2$$

where $\xi$ and $\eta$ are the North and East position respectively.

At each sampling instants only the current optimum control signal is sent to the plant while all the future ones are neglected

$$u^*(t) = \overline{u}^*(\tau, y(t), T_p), \quad \tau \in [t, t + \delta]$$

where $\delta$ is the sampling time. This process is repeated at the next cycle.

In order to reduce the optimization problem defined in equation 6 to a linear quadratic optimization problem, the output predicted trajectory is expressed as a linear combination of variations from a nominal trajectory. The nominal trajectory is obtained holding the previous input $u(t-\delta)$ over the whole horizon $T_p$. This is also called the free response: the response of the plant when the previous input is held constant over the prediction horizon. The nominal input

$$\overline{u}_n(\tau) = u(t-\delta), \quad \forall \tau \in [t, t + T_p]$$

when applied to the plant model produces the nominal output trajectory

$$\overline{y}_n(\tau) \quad \forall \tau \in [t, t + T_p]$$

A set of polynomial basis functions, $b_i(\tau)$, are then added to the nominal input:

$$\overline{u}_k(\tau) = \overline{u}_n(\tau) + b_i(\tau), \quad \forall \tau \in [t, t + T_p]$$

Different basis functions can be used to perturb the nominal input. The stability and the convergence of the MPC control problem is strictly related to the type and the perturbation level of basis function selected. The most common basis functions are step input equally or logarithmically spaced over the prediction horizon. Although these basis gave good results we found that Laguerre orthonormal basis functions helped to form a numerically well posed LQ problem. The first few Laguerre polynomials are

$$L_0(\tau) = 1$$

$$L_1(\tau) = -\tau + 1$$

$$L_2(\tau) = (\tau^2 - 4\tau + 2)/2$$

$$L_3(\tau) = (-\tau^3 + 9\tau^2 - 18\tau + 6)/6$$

The results presented in this paper use a 12th order Laguerre polynomial basis. The response of the plant to the $i^{th}$ basis function over the prediction horizon is

$$\overline{y}_i(\tau) \quad \forall \tau \in [t, t + T_p]$$

The difference between the $i^{th}$ basis output and the nominal output is

$$\varepsilon_i = \overline{y}_i(\tau) - \overline{y}_n(\tau) \quad \forall \tau \in [t, t + T_p]$$

Collecting all the basis input and output deviation from the nominal in matrix form, we have:

$$B = \begin{bmatrix} b_1 & b_2 & \ldots & b_N \end{bmatrix}$$

$$E = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \ldots & \varepsilon_N \end{bmatrix}$$

The output of the plant can be expressed as the sum of the nominal output and a linear combination of the basis output deviation from the nominal:

$$\overline{y}(\tau) = \overline{y}_n(\tau) + E\alpha \quad \forall \tau \in [t, t + T_p]$$

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}$$

The cost function of the MPC control problem can then be expressed in the following form

$$J = (\overline{y}(\tau) - y_\xi(\tau))^T Q (\overline{y}(\tau) - y_\xi(\tau))$$

$$= (\overline{y}_n(\tau) + E\alpha - y_\xi(\tau))^T Q (\overline{y}_n(\tau) + E\alpha - y_\xi(\tau))$$

$$= (E\alpha - (y_\xi(\tau) - \overline{y}_n(\tau)))^T Q (E\alpha - (y_\xi(\tau) - \overline{y}_n(\tau)))$$

Where $\varepsilon_\xi(\tau)$ is the difference between the reference trajectory and the free response. In our application the output is already the distance square from the reference trajectory and so $y_\xi(\tau)$ is equal to zero. The MPC is so reduced to a constrained LQ problem in alpha.

$$\min_{\alpha} J(\alpha) = \min_{\alpha} \frac{1}{2} \alpha^T H\alpha - F^T\alpha$$

where $H$ and $F$ can be easily calculated from equation 15.

At this point we assume a linear relationship between the basis input ($B$) and output ($E$). The
optimum $\alpha^*$ is used to calculate the optimum input at time $t$:

$$\hat{u}^*(t) = \hat{u}_n(t) + B(t)\alpha^*$$  \(16\)

As we mentioned earlier the optimum input is applied to the plant and the process is repeated at the next sampling time.

IV. APPLICATION RESULTS

The algorithms described in the previous section were implemented in C and C++, and integrated in a Simulink simulation of the MTV test vehicle. Fig. 2 shows the top level diagram of the Flight Control System Simulink subsystem. The Scheduler calls the various software tasks at their appropriate rates. Considering the slow dynamic response capability of the vehicle, the D* algorithm is called at 0.1 Hz, the B-spline/MPC controller at 0.5 Hz, and the inner loop autopilot at 25 Hz.

One of the key points in the design of the MPC controller is the definition of the prediction horizon. The horizon needs to be long enough to capture the slow dynamics that are relevant in the calculation of the cost function. The cost function is calculated based on the distance from a reference trajectory defined as North and East positions. The vehicle 90° turning time is a function of speed and it is of the order of 1 minute. Therefore, in order to track 90° turns a prediction horizon of at least 1 minute is necessary to formulate a well posed MPC problem.

The MPC controller uses a 2 Hz discrete model of the plant to predict the output trajectory. With one input control variable, the heading rate, and a prediction horizon of one minute the MPC optimum control problem has a search space of 120 variables (if no blocking is applied to the input). Fortunately the basis function approach reduces the size of the problem from 120 down to 12 independent bases, drastically reducing the complexity and the computation time.

The slow turning capability of the vehicle justifies the decision to run the D* algorithm at 0.1Hz. There is no benefit in updating the plan faster than 0.1 Hz because of the limited capability of the vehicle to avoid any obstacles that appear in close range.

In the simulation the MTV vehicle operates between 2 and 7 knots at a constant depth of 20 ft. The simulations were conducted using a map and bathymetry data of the Narragansett Bay.

An example of replanning in order to avoid a moving obstacle is shown in Fig. 10. The obstacle is moving North along the 2500 m East meridian. Several plans are calculated along way to generate alternate paths around the moving obstacle. Initially the vehicle tries a path in front of the obstacle, then due to its lower relative velocity, the D* algorithm switches to a path behind the obstacle.

Fig. 11 shows the trajectory followed by the MTV vehicle during a mission with 4 legs. The first leg brings the vehicle from the Start position to point A with a required heading of 180°, the vehicle next proceeds to points B (required heading 270°), C (225°), and D (180°) before returning to the End point. The circle around point C is necessary to achieve the 225° heading. The vectors in Fig. 11 represent the direction of a 3 knot current. A perfect measurement of the current is used in the MPC to calculate the optimum steering commands. Finally Fig. 12 shows the reference trajectory generated by the B-Spline algorithm to follow the waypoints out of a narrow inlet.
VI. CONCLUSIONS

This paper presents the results of a feasibility study for on-board path planning and vehicle control algorithms for the Manta Test Vehicle. The path planner, operating at 0.1 Hz, calculates waypoints that steer clear of obstacles in the environment. An inner-loop autopilot, running at a faster rate, generates vehicle commands based on MPC and spline-generation algorithms. Together, these approaches demonstrate the capability for real-time avoidance of moving obstacles, subject to the dynamic constraints of the test vehicle.

Future work involves exercising these algorithms under Draper Laboratory’s high fidelity C simulation framework. The path planning algorithm will also be augmented with a method for determining a minimum distance path that meets a given probability measure for successful traversal.

Acknowledgements

The authors wish to acknowledge current sponsorship of this work by the Office of Naval Research, under the UUV Project, Autonomous Operations Future Naval Capability (AOFNC) program.

References