16th Mediterranean Conference on Control and Automation
Congress Centre, Ajaccio, France
June 25-27, 2008

AUV Rendezvous Trajectories Generation for Underwater Recovery

Oleg A. Yakimenko, Douglas P. Horner, and Douglas G. Pratt, Jr.

Abstract — **For many years, autonomous underwater vehicles (AUVs) have been developed and employed for a myriad of tasks. Their ability to accurately collect and monitor oceanic conditions makes them a valuable asset for a variety of naval missions. Deploying and recovering AUVs, however, is currently largely limited to surface vessels or swimmers. The purpose of this paper is to demonstrate that by using a mathematical technique called a direct method of calculus of variations, it is possible for an AUV to autonomously compute and execute a trajectory that will allow for recovery by a submerged mobile recovery system (another AUV, submarine, etc.). The algorithm ensures that a smooth trajectory is produced that, while not traditionally optimal, is realistic and still close to the optimal solution. Also, using this technique allows the trajectory to be computed very rapidly allowing it to be recomputed every couple of seconds to accommodate sudden changes, possible adjustments and different disturbances, and therefore to be used in the real life.**

Keywords: Marine control, Real-time control, Optimization, Unmanned systems.

I. Introduction

Autonomous Underwater Vehicles (AUVs) have been of great interest to the United States Navy for quite some time. This interest began in 1994 with the Navy Unmanned Underwater Vehicles (UUV) Program Plan which promoted research and development for the employment of AUVs from submarines for mine warfare (MIW) and tactical oceanography. Since 1994 AUV concepts of operations (CONOPS) have been proposed through the UUV Master Plan, the Small UUV Strategic Plan, and Sea Power 21 [1,2]. Further mission areas may include Intelligence, Surveillance, and Reconnaissance (ISR), Anti-Submarine Warfare (ASW), Communication/Navigation Network Node (CN3), payload delivery, Information Operations (IO), Time Critical Strike (TCS), barrier patrol, and sea base support [1]. It has been over 13 years since the original concept of using AUVs was proposed and the CONOPS for AUV employment continues to increase.

Currently there are limited options to autonomously launch and recover AUVs from surface vessels and submarines. The ability to accomplish this will dramatically increase the utility of AUVs. To do so, the AUVs must have the autonomy necessary to plan and execute non-linear trajectories both to and away from the supporting vessel.

Recently there has been a demonstration of launching and recovering of a larger diameter AUV through the torpedo tube [3]. Instead of recovering to a docking station aft of the sail the recovery path would be to the recovery arm deployed out of the torpedo tube. That said, in the future there may be many AUVs deployed from the submarine, in that case deploying and recovering from the torpedo tube may not be practical.

Also, there has been recent work on AUV to AUV rendezvous for enabling high speed communication [4]. This work uses optimal control theory for calculating time and energy optimal solutions for the rendezvous path. Unfortunately, indirect method solutions cannot be computed in real-time (if at all), and use very simplified models, so they are not practical and flexible enough for a real-time implementation on an AUV except under a specific, relatively narrow set of conditions.

In a real life situation, however, the rendezvous would be unlikely to have ideal conditions. The supporting vessel may need to maneuver to avoid a collision, currents may change the approach geometry, speed adjustments may need to be made, etc. All these changes could obviously jeopardize an ideal time variant solution. Therefore, a method that is not time-variant must be pursued. While not necessarily optimal in the classical control theory sense, such a solution should be feasible and good enough to allow for autonomous AUV recovery, while still taking into consideration the factors of optimization.

This paper deals with employing the direct method of calculus of variations to generate rendezvous trajectories in faster than a real-time scale. That means that the CPU time should be of the order of 1% of the maneuver time. Direct methods introduced in dynamic optimization of the trajectories of aerospace vehicles in the 1950's were a very robust way of controlling a vehicle. It assured all boundary conditions and possible dynamic constraints were satisfied, and provided a smooth path for the entire trajectory using only a few varied parameters. Similarly, this methodology can satisfy both time and speed constraints for the case of AUV recovery fairly easily, while providing a near-optimal real-time solution.

The paper is organized as follows. Section II introduces the general scenario for the development of a path-planning methodology, which should generate trajectories for the recovery of an AUV to a mobile docking station [5]. Section III addresses the AUV model needed to develop a controller. Section IV introduces the key aspects of the proposed approach to compute rendezvous trajectories and explains the factors affecting the shape of the path. Finally, Section V presents some simulation results. The paper ends with

Manuscript received December 31, 2007.

O. A. Yakimenko is with the Naval Postgraduate School, Monterey, CA 93943-5107 USA (phone: 831-656-2826; fax: 831-656-3125; e-mail: oayakime@nps.edu).

D. P. Horner and D. G. Pratt, Jr. are with the NPS as well.

978-1-4244-2505-1/08/$20.00 ©2008 IEEE

conclusions.

II. Problem Formulation

This paper addresses a general scenario for autonomous recovery of the AUV by a mobile underwater recovery system (MURS), which can be another AUV, docking station towed from a surface ship, submarine, etc. It is assumed that this mission will proceed in a several stages as follows:

- the AUV has completed its mission and returns to a predetermined loitering point at a predetermined time;
- the MURS sends a message to the AUV suggesting a rendezvous point (area) and time;
- the AUV plans a trajectory to rendezvous with the MURS at a given position and time and sends an acknowledgment back to the MURS;
- the AUV executes the plan and recovers to the MURS.
- as the AUV gets close to the MURS, final navigation to the recovery platform is accomplished through a homing transponder.

With this recovery methodology, there are a couple of points worth expanding upon. First, once the AUV has returned to the loitering point, the MURS must be in an area to ensure the best chance of communicating with and recovering the vehicle. In this paper we assume the MURS will maintain a rectangular racetrack (however, from the algorithmic standpoint it does not really matter, so that a circular or any other track could be used instead).

Second, once the vehicle has planned and started to execute the trajectory to the MURS, the trajectory must be updatable to handle disturbances (unmodeled dynamics, currents) and different unforeseen events. These events include the cases when the MURS maneuvers inadvertently or the AUV must conduct reactive obstacle avoidance during the execution of the rendezvous path.

Third, it is assumed that the preferred method for recovering the AUV is for it to approach from the port or starboard aft quarter of MURS and maneuver to the final recovery location. The trajectory of the vehicle must be such that it avoids running the vehicle into the control or propulsion surfaces while the vehicle makes its approach to the recovery device.

Fourth, it is envisioned that a signal can be transmitted to AUV that includes some basic parameters, such as position, course, depth, and rendezvous time, so that the AUV could autonomously plan a path to rendezvous with the MURS for recovery.

In this rendezvous scenario, the MURS would establish a race track, which allows it to travel back and forth along two long track legs (see Fig.1). These legs are needed to allow sufficient time to contact the AUV (which is assumed to be in its holding pattern somewhere within communication range) and allow it to transit from its holding pattern to the rendezvous point. The proposed sequence of events is to have the MURS (at position 1 on Fig.1) signal the AUV (at position 2) commanding it to proceed to a rendezvous point by a certain time. The AUV computes the trajectory and acknowledges or denies the command (stage A on Fig.1). A denial would correspond to a violation of some constraint with a request to order another point or a different time to rendezvous. The final point of the trajectory is located approximately where the docking station would be located on the MURS in a given time. Knowing the geometry of the MURS allows an avoidance area to be constructed that corresponds to the aft control surfaces and the screw. The trajectory needs to avoid this area. Once agreed, both the AUV and the MURS proceed to position 3 for rendezvous (stage B) and by position 4 the recovery operation (stage C) is complete.

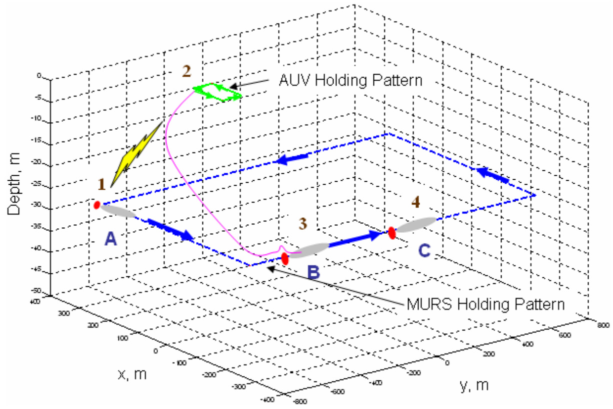

Fig. 1. Proposed rendezvous scenario.

Once again, the explored rendezvous scenario assumes three stages: communication (A), execution (B), and recovery (C), respectively. From the trajectory generation standpoint we are primarily concerned with optimizing the path that would bring the AUV from its current position (point 2) to a certain rendezvous state (point 3) in the preset (handshaked with the MURS) time T_r, while obeying all possible real-life constraints and avoiding MURS fins/screw area.

III. Vehicle Model

The Autonomous Vehicles Lab at the Naval Postgraduate School operates with several vehicles, including the REMUS AUV. The REMUS vehicle is commercially produced by Hydroid, Inc in Pocasset, MA (www.hydroid.com). A variant of the REMUS vehicle is currently used by the U.S. Navy for littoral MIW. It has a proven and long standing employment history within the U.S. Navy and was successfully used in support of Operation Iraqi Freedom (OIF) for Mine Countermeasures (MCM) in 2002. It is employed by several other Navies including the United Kingdom, Australia, and Germany. Since it is commercially produced, many of the features desired by the Navy are either already available or currently in development.

There are three model options for the REMUS vehicle itself, the 100, 600, and 6000. The differences are mainly size, operating depth, speed, and sensor packages. The model employed in OIF and owned by NPS is the model 100. This vehicle is small and perfectly suited for many operations. It is 1.6*m* (63*in*) long, 0.19*m* (7.5*in*) in diameter, and weighs only 36.3*kg* (80*lbs*) in air. It has an operational speed of 1.54*m/s* (3*kn*) allowing 22 hours of operation time or 8 hours at the maximum speed of 2.57*m/s* (5*kn*). The maximum operating depth of 100*m* (328*ft*) allows it to be invaluable in a littoral environment.

Woods Hole Oceanographic Institute has developed a prototype docking system shown in Fig.2. REMUS can use Ultra Short Baseline (USBL) navigation to locate and transit to a docking station where it may then be captured and secured. Once the vehicle is in place, a connection can be made through which data may be transferred to or from the vehicle, while its batteries recharged.

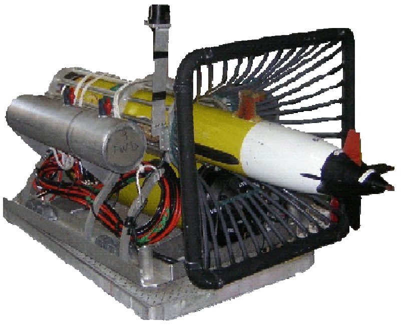

Fig. 2. Prototype of REMUS docking station [6,7].

The complete 6DoF model of the REMUS AUV to support this study and test the proposed rendezvous algorithms has been developed [8,9]. It accounts for standard assumptions (vehicle behaves as a rigid body, the Earth's rotation is negligible, all of the forces that act on the vehicle have either inertial or gravitational origin) and include linearized dynamic differential equations for surge (u), sway (v), heave (w) linear velocities, and roll (p), pitch (q), yaw (r) angular velocities. These equations are fairly common and their development is omitted here. These dynamic equations are augmented with six kinematic equations. Specifically, with three equations that relate local tangent plane (North-East-Down) coordinates of AUV's center of gravity (x, y and z) to the components of the velocity vector expressed in the body frame $\{b\}$:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = {}^{u}_{b}R \begin{bmatrix} U_0 \\ v \\ w \end{bmatrix}. \qquad (1)$$

The rotation matrix ${}^{u}_{b}R$ is given by

$${}^{u}_{b}R = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi & \cos\psi\sin\theta \\ \sin\psi\cos\theta & \cos\psi & \sin\psi\sin\theta \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \qquad (2)$$

using pitch angle θ and yaw angle ψ (bank angle φ is small and can be neglected). Moreover, for simplicity we will further reduce the matrix (2) to

$${}^{u}_{b}R = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

assuming relatively small pitch angles typical for AUVs ($\theta \le 30°$). Equation (1) implies that in our specific application we have no control over a surge component of the speed, assuming it to be constant, U_0 . In case the currents are known, they can be added to the right-hand side of equation (1) to get proper velocity components in the inertial frame.

Normally, stabilization would be the first priority for developing a controller, but the REMUS vehicle comes with a primary controller that takes care of it already installed. This primary controller serves several functions among which are stabilization and directly controlling the control surfaces and propeller. The NPS REMUS vehicle also has the optional Remote Control Protocol (RCP) installed. The ASCII text serial protocol permits a secondary CPU to take overriding control of the vehicle. This is normally accomplished via the higher-level inputs sent to the primary controller. These inputs may include heading Ψ (or yaw rate $\dot{\psi}$), depth z (or flight path angle γ), and speed (V). This secondary controller is useful when coupled with a sensory system that can provide information for greater autonomy within the AUV. An example is using forward looking sonar to enable reactive obstacle avoidance [9-12].

Summarizing, for the purposes of this paper we assume that the primary and secondary controllers discussed above are combined into one autopilot that together with the AUV itself forms a stable enhanced plant shown on Fig.3. By making this assumption, we remove both the need to stabilize the system and to generate actual commands for the control surfaces and rudder. Instead, to control the vehicle we only need to generate a set of signals consisting of yaw rate, flight path angle, and speed that are the only inputs to the autopilot-enhanced system necessary for navigating from one position to another. (In what follows we even reduce a set of signals to only two, excluding speed.)

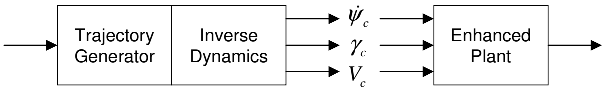

Fig. 3. Incorporation of Trajectory Generator into the REMUS control scheme.

IV. Rapid Prototyping of Rendezvous Trajectories

The proposed real-time trajectory generator block shown in Fig.3 employs the modification of the direct method of calculus of variations originally developed for aircraft in the mid 60's [13]. In one of its versions the entire trajectory was represented as a combination of three third-order polynomials for each of the three coordinates (and speed) developed in the virtual domain of some abstract argument τ [14]. Later, it was shown that applying higher-order polynomials allows satisfying higher-order derivatives of coordinates at the terminal points making it possible to incorporate this approach onboard the vehicle more effectively [15]. Married together, the two approaches ([14] and [15]) yield a very effective (from the computational standpoint) and robust approach for generating feasible ready-to-track short-term maneuvers in real-time [16,17]. This latter approach has already been applied to generate trajectories for different vehicles including AUVs [10,18].

Skipping mathematics that has been already addressed for instance in [16] and [18], it can be noted that the entire 3D trajectory is represented as three seventh-order polynomials, depending on the vectors of preset initial and final boundary conditions (including up to the second-order derivative of coordinates) and a vector of varied parameters $\mathbf{\Omega}$ as follows:

$$\begin{aligned} x(\tau) &= P_x(\mathbf{X}_0, \mathbf{X}_f, \mathbf{\Omega}, \tau) \\ y(\tau) &= P_y(\mathbf{X}_0, \mathbf{X}_f, \mathbf{\Omega}, \tau) \\ z(\tau) &= P_z(\mathbf{X}_0, \mathbf{X}_f, \mathbf{\Omega}, \tau) \end{aligned} \tag{4}$$

(Note that these polynomials are developed in the virtual domain and not in the time domain, allowing for independent optimization of the speed profile.)

Vectors $\mathbf{X}_0$ and $\mathbf{X}_f$ are composed of: a) the three components of the current initial and desired final position of the AUV, b) three components of the current initial and desired final velocity of the AUV, and c) the three components of the current initial and desired final acceleration of the AUV. Since no radical maneuvers at the terminal point are desired, the acceleration components at the final point (proportional to the second-order derivatives of coordinates) are all set to be zero.

For the specific problem of generation the AUV rendezvous trajectories, the vector of varied parameters $\mathbf{\Omega}$ includes the value of the arc length τ_f along with the third-order derivatives of coordinates at the terminal points (erk), in order to adjust the trajectory to meet all constraints.

As shown on Fig.3, the trajectory generator block also includes inversing vehicle's dynamics to develop necessary controls and account for all constraints. Once the candidate trajectory (coefficients of polynomials (4)) is computed, inverse dynamics is then used to calculate other states and the required controls at each point on the path. The values produced by the trajectory generation algorithm and inverse dynamics are then used to compute the performance index and estimate the degree of possible violation of any constraint. Using these data the varied parameters will then be adjusted accordingly to achieve the minimum of the performance index while satisfying all constraints.

What these constraints are? For any vehicle, the control surfaces can only move so far and it can only go so fast. Keeping in mind the block-diagram of Fig.3, for the REMUS vehicle this translates to constraints being approximately equal to 10°/*s* for the yaw rate, $\dot{\psi}$, and 20° for flight path angle, *γ*. Since there is an area that we do not want the trajectory to go through, another constraint is also added, so that the computed trajectory remains outside of a given volume. This volume is most easily represented as a sphere with a radius of about 5 meters and having its center positioned, so that the entire avoidance area is contained within it. Finally, and not immediately obvious, is the fact that some trajectories may attempt to take the vehicle out of the water or below the sea floor. Since neither of these conditions is feasible, the range of depth allowed for the trajectory must be limited.

The performance index may have several components with the major one being proportional to the squared difference between the computed time of the rendezvous maneuver t_f and predetermined rendezvous time T_r, i.e. $(t_f - T_r)^2$. Other terms might account for minimization of control efforts necessary to bring the AUV to the rendezvous point, for instance $\int_0^{t_f}\left[\left(\Psi(t) - \tan^{-1}\frac{y_f - y_0}{x_f - x_0}\right)^2 + \gamma(t)^2\right]dt$.

Before proceeding with the derivation of inverse dynamics, it is important to remember that the trajectory is computed along a virtual arc and not in the time domain. This means that there must be some way to convert from the virtual arc domain, *τ*, and the time domain, *t*. This conversion is given by

$$\lambda = \frac{d\tau}{dt}, \tag{5}$$

where *λ* is the so-called speed factor [14]. The discrete analogue of (5) is

$$\lambda_j = \Delta\tau\Delta t_{j-1}^{-1}, \; j = 2,...,N, \tag{6}$$

where

$$\Delta\tau = \tau_f (N-1)^{-1}, \tag{7}$$

and *N* is the number of increments that the arc length, τ_f, is broken into during the numerical procedure.

Equation (6) indicates that λ_j is a function of the change in *τ* divided by the instantaneous change in time. The subscript on time indicates that this time step $\Delta t_{j-1} = t_j - t_{j-1}$ is not constant and must also be computed. Intuitively, this calculation should, and does involve dividing the distance between two points along the arc by the speed:

$$\Delta t_{j-1} = \frac{\sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2 + (z_j - z_{j-1})^2}}{\sqrt{U_0^2 + v_{j-1}^2 + w_{j-1}^2}}. \tag{8}$$

Now let us recall our system dynamics given by equation (1) that should be rewritten in the virtual domain as

$$\lambda_j \begin{bmatrix} x'_j \\ y'_j \\ z'_j \end{bmatrix} = {}^u_bR(\psi_j) \begin{bmatrix} U_0 \\ v_j \\ w_j \end{bmatrix}. \tag{9}$$

In order to compute sway and heave velocities needed in (8) we need to invert system (9) with respect to these two plus unknown yaw angle.

We can represent this inversion in the matrix form as

$$\begin{bmatrix} U_0 \\ v_j \\ w_j \end{bmatrix} = \lambda_j \begin{bmatrix} \cos\psi_j & \sin\psi_j & 0 \\ -\sin\psi_j & \cos\psi_j & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_j' \\ y_j' \\ z_j' \end{bmatrix} \qquad (10)$$

or in the scalar form as

$$U_0 = \lambda(x_j' \cos\psi_j + y_j' \sin\psi_j)\,, \qquad (11)$$

$$v_j = \lambda_j(-x_j' \sin\psi_j + y_j' \cos\psi_j)\,, \qquad (12)$$

$$w_j = \lambda_j z_j'\,. \qquad (13)$$

While the equation (13) is readily available to compute the next time step using (8) (the derivatives of x, y and z are given analytically by differentiating equations (4) with respect to τ), equations (11)-(12) need to be excluded of an unknown yaw angle ψ, which yields

$$v_j = \sqrt{\lambda_j^2(x_j'^2 + y_j'^2) - U_0^2}\,. \qquad (14)$$

Now, in order to begin using equations (13) and (14) in (8) the initial value of λ must be assessed. Since equation (5) is literally a change in arc length per unit time, it is feasible to assume that the initial value of λ is equal to the initial speed of the vehicle. It is true in case of the virtual arc length τ_f having the order of the physical path length s_f. After each iteration we may readjust it according to

$$\lambda_1 = U_0 \frac{\tau_f}{s_f}\,. \qquad (15)$$

Equations (11) and (12) can also be resolved to yield ψ

$$\psi_j = \tan^{-1}\frac{y_j'}{x_j'} - \tan^{-1}\frac{v_j}{U_0}\,, \qquad (16)$$

and therefore a command yaw rate

$$\dot{\psi}_c(t_j) = (\psi_j - \psi_{j-1})\Delta t_{j-1}^{-1}\,. \qquad (17)$$

The command flight path angle is defined as

$$\gamma_c(t_j) = \tan^{-1}\frac{-z_j'}{\sqrt{x_j'^2 + y_j'^2}}\,. \qquad (18)$$

V. Simulation Results

This section presents an example of the trajectory generation using the approach discussed above. The goal is to be able to compute a rendezvous trajectory from any point on the AUV holding pattern to any point on the MURS holding pattern as shown on Fig.4 (for stochastic simulation the circular race tracks were employed).

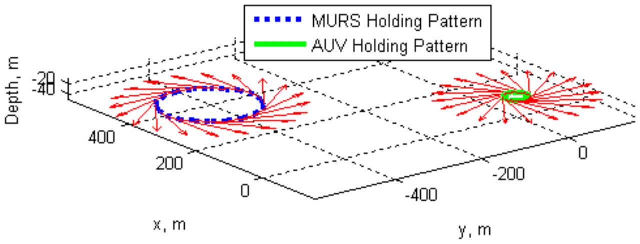

Fig. 4. Manifold of initial and final conditions.

Specifically, using the scenario stated in Section II, a MURS is moving due east at 1*m/s* (1.94*kn*) with the docking station at a depth of 15m. A REMUS vehicle is located 800 meters away. The MURS wishes to conduct rendezvous operation T_r minutes later and sends the corresponding information in to REMUS. This information includes the proposed final position, x_f, y_f, z_f, rendezvous course, speed and time.

With the optimization procedure an initial guess is made regarding the virtual arc length of the trajectory and the required components of the initial and final jerks. It takes only a few seconds (in the Mathworks MATLAB/Simulink development environment) to optimize the trajectory to the final one satisfying all constraints and reaching the rendezvous point in exactly preset time (MATLAB's *fminsearch* function was used to minimize the performance index augmented with weighted penalties for constrains violation).

Several generated trajectories meeting the desired objectives for the chosen scenario are shown in Fig.5 along with an obstacle on the way to MURS the AUV needs to avoid. These trajectories differ by the arrival time T_r.

While handshaking with the MURS, the AUV determines whether suggested T_r is feasible. To this end, among four shown trajectories the first one, with T_r=450*s*, happens to be infeasible (the constraints on controls are violated). The solution of the minimum-time problem for this scenario yielded 488*s* as a soonest possible rendezvous time.

Three other trajectories on Fig.5 are feasible. That means that the boundary conditions are met (by construction) and all constraints including obstacle avoidance are satisfied (via optimization). As an example, Fig.6 depicts time histories of REMUS vehicle's control parameters of yaw rate $\dot{\psi}_c$ and flight path angle γ_c for the trajectory with T_r=600*s*. The third plot shows the speed time history.

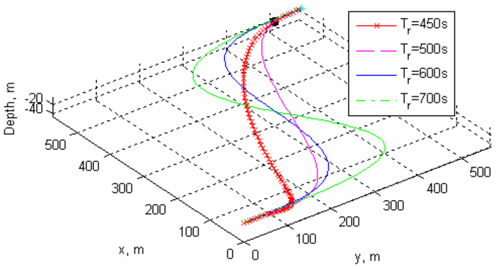

Fig. 5. Examples of rendezvous trajectories.

The stochastic simulation for manifolds shown on Fig.4 showed that in all of those cases the rendezvous can take place if T_r is greater than a certain value. Furthermore, they show that regardless of the initial guess the minimization of the performance index ensures that a smooth, realizable trajectory is calculated just in a few seconds (conversion to the executable file rather than using interpretative programming language code would reduce this time even further).

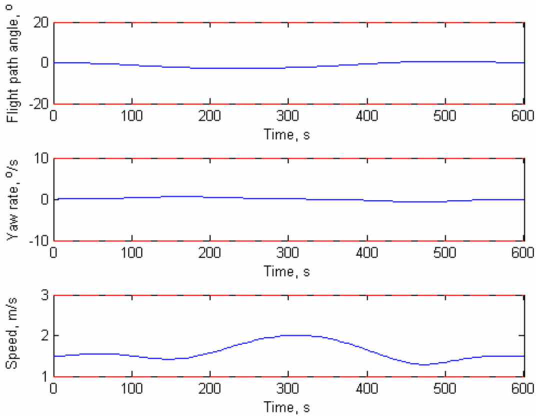

Fig.6. Constrained vehicle parameters for T_f=600s.

VI. Conclusion

Based on the simulations it can be stated that using a direct method to calculating rendezvous trajectories results in a smooth, realizable path. Constraints can be inserted to ensure that not only will vehicle parameters not be violated, but that we may also define limits on the trajectory itself. This demonstrates and supports the theoretical feasibility of using the direct method for underwater recovery of AUVs. It is expected that further exploration for using this technique will include more computer simulations incorporating the models of primary and secondary controllers working together subject to disturbances and experimentation using the NPS REMUS vehicle at the Naval Postgraduate School.

References

[1] "The Navy Unmanned Undersea Vehicle (UUV) Master Plan," Department of the Navy, November 9th, 2004. Available: www.chinfo.navy.mil/navpalib/technology/uuvmp.pdf

[2] V. Clark, "Sea Power 21: Projecting Decisive Joint Capabilities," *United States Naval Institute Proceedings*, October 2002. *http://www.usni.org/magazines/proceedings/archive/month.asp?ID=114.*

[3] "Boeing's BLQ-10 LMRS: A Sub-Recoverable UUV?" *Defense Industry Daily*, November 28th, 2007. http://www.defenseindustrydaily.com/boeings-blq-11-lmrs-a-sub-recoverable-uuv-04319/

[4] J. W. Nicholson, "Autonomous Optimal Rendezvous of Underwater Vehicles", MS Thesis, NPS, Monterey, CA, September, 2004.

[5] D. Pratt, "AUV Rendezvous Using Real-Time Near Optimal Trajectories for Submarine Recovery", MS Thesis, NPS, Monterey, CA, December 2007.

[6] B. Allen, T. Austin, N. Forrester, R. Goldsborough, A. Kukulya, G. Packard, M. Purcell, and R. Stokey, "Autonomous Docking Demonstrations with Enhanced REMUS Technology," in *Proc. Oceans'06 MTS/IEEE Conference*, Boston, MA, September 18-21, 2006. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4099107

[7] A. Kukulya, "REMUS 100," Oceanographic Systems Laboratory of the Woods Hole Oceanographic Institution, MA, October, 2007. http://www.whoi.edu/sbl/liteSite.do?litesiteid=7212&articleId=11154.

[8] T. Prestero, "Verification of the Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle", MS Thesis, MIT, MA, September, 2001.

[9] L. Fodrea, "Obstacle Avoidance Control for the REMUS Underwater Autonomous Vehicle", MS Thesis, NPS, Monterey, CA, December, 2002.

[10] D. Horner, and O. Yakimenko, "Recent Developments for an Obstacle Avoidance System for a Small AUV, " in *Proc. IFAC Conference on Control Applications in Marine Systems*, Bol, Croatia, September 19-21, 2007.

[11] T. Furukawa, "Reactive Obstacle Avoidance for the REMUS Underwater Autonomous Vehicle Using a Forward Looking Sonar", MS Thesis, NPS, Monterey, CA, June 2006.

[12] D. P. Horner, A. J. Healey, and S. K. Kragelund, "AUV Experiments in Obstacle Avoidance," in *Proc. Oceans'05 MTS/IEEE Conference*, Washington, D.C., September 18-23, 2005.

[13] V. T. Taranenko, *Experience of Employing the Ritz's, Poincare's, and Lyapunov's Methods for Solving the Problems of Flight Dynamics*, Air Force Engineering Academy (AFEA), Moscow, 1968 (in Russian).

[14] V. T. Taranenko, and V. G. Momdzhi, *Direct Method of Calculus of Variations in Boundary Problems of Flight Dynamics*, Maschinostroenie, Moscow, 1986 (in Russian).

[15] A. I. Neljubov, *Mathematical Methods of Calculation of Combat, Takeoff/Climb and Landing Approach Maneuvers for an Aircraft with 2D Thrust Vectoring*, AFEA, Moscow, 1986 (in Russian).

[16] O. A. Yakimenko, "Direct Methods for Rapid Prototyping of Near-Optimal Aircraft Trajectories," *Journal of Guidance, Control, and Dynamics*, vol.23, no.5, 2000, pp.865-875.

[17] O. A. Yakimenko, Y. Xu, and G. Basset, "Computing Short-Time Aircraft Maneuvers Using Direct Methods," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, HI, August 18-23, 2008.

[18] O. A. Yakimenko, "Real-Time Computation of Spatial and Flat Obstacle Avoidance Trajectories for UUVs," in *Proc. 2nd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV'08)*, Killaloe, Ireland, April 8-10, 2008.