# ARRAY SHAPE DETERMINATION WITH A REAL-TIME MODEL

A.A. Ruffa

*Naval Undersea Warfare Center Detachment, ASW Systems Department, New London, CT 06320, U.S.A.*

**Abstract** A shape estimation scheme based on precomputed frequency-domain solutions of array motion is presented. The boundary conditions supporting each solution are discrete Fourier transform (DFT) components of sensor measurements, which are updated in real time. A prediction of array shape in the time domain is obtained by performing an inverse DFT on the set of frequency-domain solutions. Extensions to the method include the use of slope inputs instead of displacements as well as inclusion of an arbitrary number of inputs to minimize the effect of measurement error. A comparison of the results with data shows good agreement, and CPU time is an order of magnitude faster than real time on the VAX 8800 (with an update rate of 0.74 sec).

## 1. Introduction

It is widely recognized that towed array shape must be addressed for future low frequency active ASW operations. There are two principal areas of concern. The first involves array shape during a gross maneuver (such as a turn), and the second involves perturbations to the ideal array shape under steady tow conditions. This paper addresses the latter problem.

Any prediction scheme intended for tactical use must be real time in nature. In addition, it must use heading sensors rather than pingers as a source of input, because the former are much more likely to be available. Finally, it must contain a scheme to minimize the introduction of (potentially large) input errors. This implies the use of a least-squares method to obtain the shape that best fits sensor measurements.

The approach taken in this paper is to determine the array shape that satisfies the physics of its motion. In the next section, the equation governing array dynamics is expressed in terms of both displacement and heading. A real-time solution to this equation and a comparison with data are then presented. Finally, a method is discussed to fit this solution to an arbitrary number of sensors.

## 2. Governing Equation

Paidoussis [1],[2] formulated the problem of describing perturbations about the ideal shape of a towed array, leading to the following partial differential equation:

$$A\,\partial^2 y/\partial x^2 + B\,\partial^2 y/\partial x\partial t + C\,\partial^2 y/\partial t^2 + D\,\partial y/\partial x + E\,\partial y/\partial t + F\,\partial^4 y/\partial x^4 = 0 \qquad (1)$$

where

$$\begin{aligned} A &= T(x) - \rho\pi a^2 U^2 \\ B &= -2\rho\pi U a^2 \\ C &= -2\rho\pi a^2 \\ D &= -\rho\pi a U^2 C_N \\ E &= -\rho\pi a U C_N \\ F &= -\pi a^4 E_y/4 \\ T(x) &= \rho\pi a U^2 C_T (L - x). \end{aligned}$$

Here U is the tow speed, a is the array radius, $\rho$ is the density of water and array, $E_y$ is the array modulus of elasticity, $C_T$ is the tangential drag coefficient, L is the distance from the towpoint to the free end, and $C_N$ equals $\alpha\, C_T$ where $0 \leq \alpha \leq 1$ [3].

Equation (1) describes the transverse displacements y (x, t) at a distance x from the towpoint in an array modeled as a neutrally buoyant circular cylinder towed through its axis (except for small transverse displacements introduced at the towpoint). Many methods have been used to solve this equation (e.g., [3],[4]). However, a numerical solution of (1) in the time domain leads to numerical diffusion [5], which has the effect of damping all but the low-frequency perturbations. For this reason, it is preferable to solve (1) numerically in the frequency domain.

If the displacements take the form $y(x, t) = Y(x, \omega)\, e^{i\omega t}$, (1) reduces to the following:

$$A\,d^2Y/dx^2 + (i\omega B + D)\,dY/dx + \left(i\omega E - \omega^2 C\right) Y + F\,d^4Y/dx^4 = 0. \qquad (2)$$

The finite difference method is used to solve (2) numerically. As boundary conditions, measured displacements $y(x_i, t)$ are specified (where $i = 1, 2$; $x_1 = L_1$; $x_2 = L_2$; $L_1$, $L_2$ are pinger locations) as well as zero slopes at these locations. Alternatively, an analytical free-end boundary condition can be used in lieu of the conditions at $x_2$ [2]:

$$d^2y/dx^2 = 0$$

$$F\, d^3y/dx^2 + -1/2\, fB\, (\partial y/\partial t + U \partial y/\partial x) = 0 \tag{3}$$

where f is a nondimensional parameter [1].

In practice, imposing conditions at $x_2$ upstream of the free end yields significantly more accurate solutions than (3). Therefore, it is necessary to make measurements at two locations to obtain an accurate shape prediction.

Differentiating (1) with respect to x makes it suitable for heading sensor (slope) input:

$$A\, \partial^2 s/\partial x^2 + B\, \partial^2 s/\partial x \partial t + C\, \partial^2 s/\partial t^2 + (D+G)\, \partial s/\partial x + E\, \partial s/\partial t + F\, \partial^4 s/\partial x^4 = 0 \tag{4}$$

where

$$s(x,t) = \partial y(x,t)/\partial x$$
$$G = -\rho \pi a C_T U^2.$$

As before, a harmonic time dependence is assumed, leading to the following:

$$A\, d^2S/dx^2 + (i\omega B + D + G)\, dS/dx + \left(i\omega E - \omega^2 C\right) S + F\, d^4S/dx^4 = 0. \tag{5}$$

Boundary conditions consist of specification of slopes $s(x_i, t)$. In addition, $\partial s/\partial x$ is set to zero at $x_i$.

### 3. A Real-Time Solution and Results

The solution to (1) in the time domain can be expressed as follows:

$$y(x,t) = \sum_{n=0}^{N-1} Y(x, \omega_n) e^{i\omega_n t} \tag{6}$$

where $\omega_n = n/(N\Delta t)$.

To speed up computation of Y (x, $\omega_n$), special frequency-domain solutions $\Psi_i$ (x, $\omega_n$) are precomputed and stored. Each of the precomputed solutions has the property $\Psi_i(x_k, \omega_n) = \delta_{ik}$, i.e., zero displacement is specified on one boundary and a unit displacement is specified on the other. Zero slopes are also specified on both boundaries. The solution Y (x, $\omega_n$) can be written as [6]

$$Y(x,\omega_n) = \sum_{i=1}^{2} b_i(\omega_n)\, \Psi_i(x,\omega_n). \tag{7}$$

Here $b_i(\omega_n)$ are the discrete Fourier transform (DFT) components of the boundary displacement inputs y ($x_i$, t). After $b_i(\omega_n)$ are calculated from N time steps, they can be updated using the following algorithm:

$$b_i(\omega_n) = b_i(\omega_n) + (1/N)\left[y(x,t) - y(x,t-N\Delta t)\right]e^{-2\pi i\omega_n t}. \tag{8}$$

Figures 1 and 2 compare array shape predictions with displacement data measured at an intermediate location between $x_1$ and $x_2$. Displacement measurements from pingers mounted in the array [7] provided boundary condition displacement inputs y ($x_i$, t). The distance between $x_1$ and $x_2$ is 55 m. Each run took approximately 15 sec of CPU time on a VAX 8800 to simulate more than 300 sec (with a time step/update rate of 0.74 sec).

The comparison, which was made 27 m from $x_1$, is generally good. The largest differences, which are on the order of measurement error, are in the high-frequency components in figure 2. Measurement error can affect the solution through the boundary conditions, and it can be present in the data used for comparison. The next section addresses the minimization of the effects of measurement error.

The methods in this section apply equally well to (4), making possible a real-time shape estimation scheme using only heading sensor (slope) input. The computation is slower, however, since displacements y (x, t) must be integrated from s (x, t).

## 4. Inclusion of Additional Sensor Input Data

A shape prediction method derived from solving the equations of array motion is limited to two sensor inputs (i.e., those used for boundary conditions). Introduction of measurement errors by one or both of the sensors can degrade the predicted shape significantly. In this section, a method is presented that minimizes the error (in a least-squares sense) between the solution to (1) or (4) and an arbitrary number of sensor inputs.

Applying the finite difference method to (2) or (5) leads to a system of equations that can be written in the form $\mathbf{A}\cdot\mathbf{X} = \mathbf{B}$, where **A** is an M by M matrix, and **X** and **B** are vectors of length M (M is the number of nodes in the finite difference scheme). The system of equations is overdetermined, however, when more than two sensor inputs are

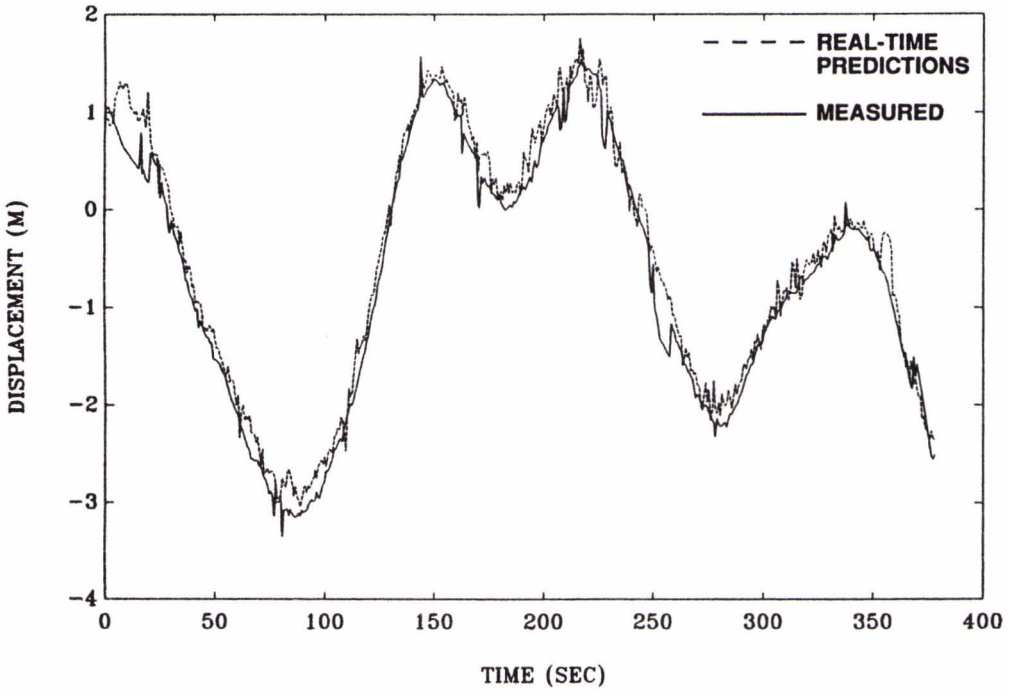


**Figure 1. Transverse Displacement as a Function of Time**

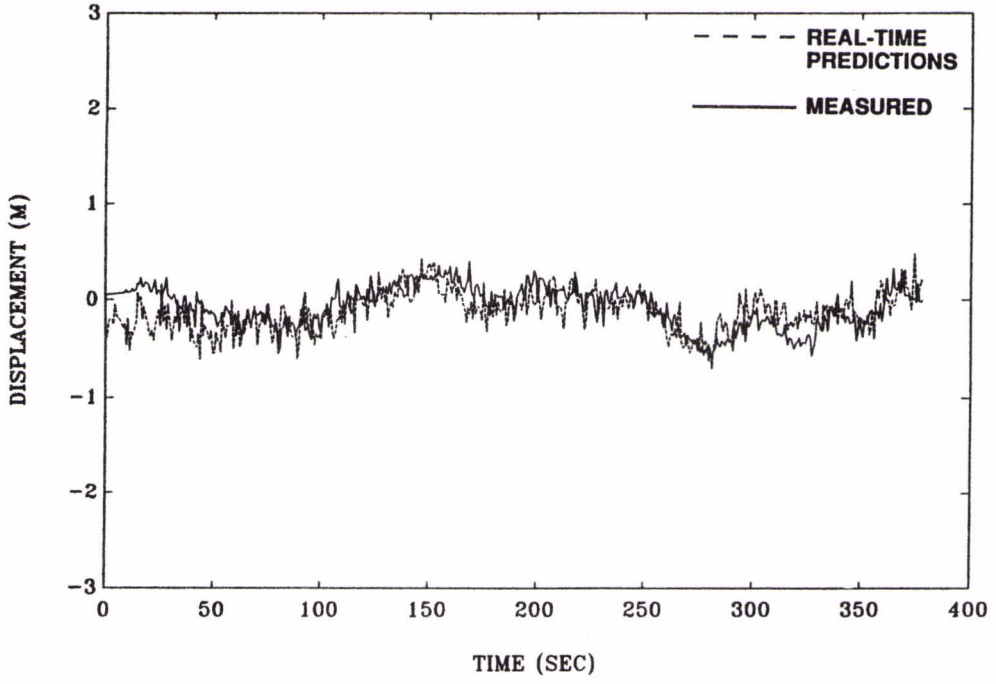


**Figure 2. Vertical Displacement as a Function of Time**

used. If there are P sensors, **A** becomes a P + M – 2 by M matrix, **X** has length M, and **B** has length P + M – 2. Such a system of equations can be "solved" using the singular value decomposition [8] by representing **A** as the product of orthogonal matrices **U** (P + M – 2 by M) and $\mathbf{V}^T$ (M by M) and a diagonal matrix **W** (M by M):

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{W} \cdot \mathbf{V}^T. \tag{9}$$

Therefore, the "solution," **X**, can be expressed as follows:

$$\mathbf{X} = \mathbf{A}^{-1} \cdot \mathbf{B} = \mathbf{V} \cdot \mathbf{W}^{-1} \cdot \mathbf{U}^T \cdot \mathbf{B}. \tag{10}$$

It can be shown [8] that (10) represents the best least-squares fit to the overdetermined system. Therefore, the solution does not necessarily pass through each measurement. Instead, the difference between the solution and each measurement is minimized in a least-squares sense. This also minimizes the effect of any sensor measurement error.

The above procedure can also be implemented in real time as a solution to either (1) or (4). For example, if there are P sensors, a best-fit solution $Y(x, \omega_n)$ to (2) is obtained by precomputing solutions $\Psi_i(x, \omega_n)$ for P cases, each corresponding to a unit input at $x_i$ (i = 1, P) and zero input for all others (i.e., $\Psi_i(x_k, \omega) = \delta_{ik}$). The solution is then

$$Y(x,\omega_n) = \sum_{i=1}^{P} b_i(\omega_n)\, \Psi_i(x,\omega_n). \tag{11}$$

As before, $b_i(\omega_n)$ is the DFT component of $y(x_i, t)$ at frequency $\omega_n$. Storage of the precomputed solutions $\Psi_i(x, \omega_n)$ allows real-time computation of $Y(x, \omega_n)$ and $y(x, t)$ using a procedure similar to that in section 3.

## 5. Conclusions

An accurate, real-time shape estimation scheme can be derived from the solution to the equations of array motion. Heading sensor (slope) input can be used instead of displacements, and the limitation of two inputs can be removed to minimize the effect of measurement error.

## Acknowledgment

This work was conducted in support of a NUWC project in Multi-Line Towed Array (MLTA) Cable Dynamics, Principal Investigator E.Y.T. Kuo (Code 3321). The sponsoring activity was the Office of Naval Technology, Submarine/Surface Ship ASW Surveillance Block of the Naval Undersea Warfare Center, Program Manager T. Goldsberry.

**References**

[1] Paidoussis, M.P. Dynamics of flexible slender cylinders in axial flow. *Journal of Fluid Mechanics*, **26**, 1966: 717-736.

[2] Paidoussis, M.P. Dynamics of cylindrical structures subjected to axial flow. *Journal of Sound and Vibration*, **29**, 1973: 365-385.

[3] Dowling, A.P. The dynamics of towed flexible cylinders, part 1: Neutrally buoyant elements. *Journal of Fluid Mechanics*, **187**, 1988: 353-571.

[4] Kuo, E.Y.T. and Ruffa, A.A. Elements of numerical system analysis for a multi-line towed array. NUWC TM 871113. Naval Undersea Warfare Center Detachment, New London, CT, U.S.A., 1986.

[5] Kuo, E.Y.T.; Ruffa, A.A.; Tober, B.M.; Charette, R.H.; and Casati, M.J. MLTA cable dynamics and acoustic performance: FY89. NUWC TM 891129. Naval Undersea Warfare Center Detachment, New London, CT, U.S.A., 1989.

[6] Ruffa, A.A. and Casati, M.J. Improvements to a real time model for towed array dynamics. NUWC TM 901165. Naval Undersea Warfare Center Detachment, New London, CT, U.S.A., 1990.

[7] Beam, J.P. Multiline towed array (MLTA) partial array sea test final report on array motion (U). NUWC TM 871185. Naval Undersea Warfare Center Detachment, New London, CT, U.S.A., 1987. (Confidential)

[8] Press, W.H.; Flannery, B.P.; Teukolsky, S.A.; and Vetterling, W.T. Numerical Recipes. Cambridge University Press, Cambridge, U.K., 1989.