Flapping Foil Computations for Station Keeping Buoy

Background

To investigate concepts for an unmoored, station-keeping buoy system that generates propulsion and electricity from ocean waves, MBARI has developed a numerical model that solves the equations of motion of one or more floating or submerged bodies that are connected by a tether with a specified tension/strain relationship. The equations of motion for rigid bodies are of course well known and the key parts of the solver are the evaluation of the forces acting on the bodies as a function of the systems state (position and velocity history of all bodies). For floating bodies, the forces of the waves acting on the body are required and routines have been developed to evaluate these forces based upon linear, invisid wave theory. For simple submerged bodies drag coefficients of known shapes can be used to evaluate the required forces. To validate the validity of the numerical model at the scales being investigated, an at-sea test has been completed that coupled a flat plate (48" diameter, 1300lbs) to a floating buoy (92" diameter, 2000lbs) via a tether that included a spring/damper element. The measured results for tension and elongation of the spring element match satisfactorily with the modeled results in a similar sea-state.

A leading concept for the station keeping buoy involves a surface expression tethered to a submerged foil that can generate lift and subsequently propulsion. In this case simple drag coefficients have limited utility due to the transient nature of the flow and we desire to investigate options for coupling the equation of motion solver to a computational fluid dynamics solution of the lifting flow of a submerged foil whose motion is driven by a connection to a surface buoy. There are one or more assumptions that will make this task simpler, but the validity of each of these is unknown at this point. This document outlines the various levels of assumptions possible and describes the resulting computational requirements for each. A final section outlines an approach to explore the various options and is meant to be a template for a statement of work to be incorporated into a contract between MBARI and Airflow Sciences, after refinement and discussion.

Equation of Motion Solver Requirements

The solver is written in the Matlab programming language and utilizes a 4th order Runge-Kutta algorithm to solve the equations of motion of one or more body in 3 dimensions (6 degrees of freedom). Any new body can be modeled by specifying the physical characteristics of the body (mass, moment of inertial, displacement, etc.) and defining functions that compute the force on the body due to the relevant effects. The code is structured so that separate functions apply separate forces, providing a simple way to add additional forcing functions for specific effects. For example, the existing code to model floating bodies includes separate functions to evaluate forces due to the bodies weight, the bodies buoyancy, and the wave forcing on the body. If one wanted to include a force due to wind loading, a new function would be created that evaluates the wind loading based upon the bodies current velocity and position.

Any given force function has access to the state history (position and velocity) of all bodies in the simulation. In most cases only the state of the body that the force acts upon is required to evaluate the force, but in cases such as tethers that connect two bodies, the state of both bodies is required. A standard interface exists for passing the bodies state to these separate functions.

To include a computational fluid dynamics solution of the flow around a submerged foil in the general case, it will be necessary for the CFD solver to accept as an input the new position and velocity of the body and return the resulting force due to the surrounding fluid. This is the most general case and has the downside that the equation of motion solver and the CFD solver must be coupled, making the CFD results meaningless outside of the particular simulation. Several simplifying assumptions relative to the specific case of modeling a submerged foil are discussed below.

As noted above, the solver is a 3 dimensional solver with 6 degrees of freedom. Practically, the investigations of interest are mostly two dimensional effects and many of the results generated to date have only included forcing for three degrees of freedom (heave, surge, and pitch), the solver continues to integrate the solution for the other modes of motion but the solution is trivially zero. In addition to making the results simpler to understand, this approach as the added benefit of avoiding the often thorny problem of representing the motion of rigid bodies in three dimensions through the use of rotation matrices, Euler angles, or quaternions.

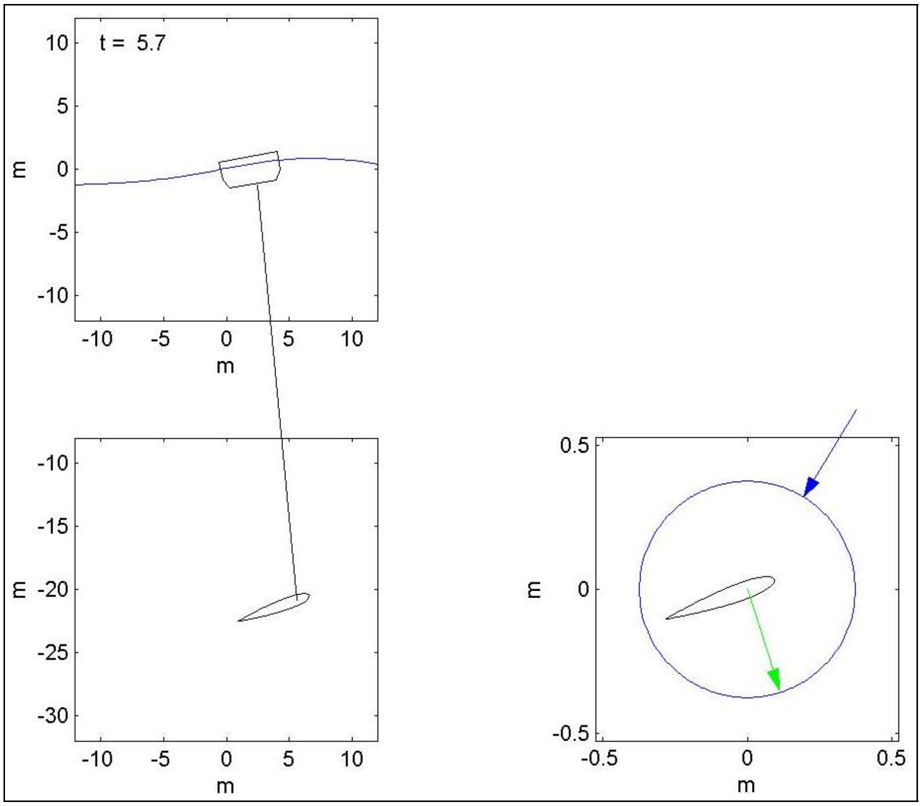

Figure 1: Illustration of submerged foil below surface buoy in waves. The surface buoy in the upper left and the foil in the lower left are enlarged for clarity, the buoy has a diameter of 92" and the foil has a chord length of 16". The blue arrow in the lower right plot shows the pitch angle and flow direction relative to a coordinate system centered on the foil, and parallel to the base coordinate system. The green arrow shows the lift/drag resultant force on the foil based upon lift/drag curves at Reynolds number $1X10^6$. An animation of the simulation can be seen at: http://www.mbari.org/staff/hamilton/Videos/FoilExample.mpg

Flapping Foil Dynamics: Lift, Drag and Pitch Control

The concept of the flapping foil propulsion relies on the motion of the buoy to create a fluid flow over the foil which has a net force in the positive direction. Although propulsion will be created by a purely heaving foil, it is inefficient. Controlling the pitch of the foil to increase the percentage of time the foil is operating in a lifting mode increases the efficiency and the net propulsive force. For the purposes of simulation, there are two possible approaches to the pitch control issue. The first is to specify a desired pitch behavior relative to the foil motion and implement a control loop in the simulation that applies a moment to drive the foil to the desired orientation. This approach is simple and has the advantage of making it possible to explore the effect of various pitch control strategies without being concerned with the particular mechanism. A more sophisticated analysis would include a solution of the pitch dynamics. In this case one would be modeling the pitch control mechanism directly and would require that the pitching moment on the foil is computed. Additionally, a suitable platform for the pitch control moment to act against must be modeled, i.e. a sled that holds the foils. Both types of analysis have a purpose but only the second requires a computation of the pitching moment on the foil due to the flow.

The lift/drag vector that results from the flow is in general time-dependent and sensitive to the past history of the foil motion and fluid flow. In this general case, the fluid flow over the foil must be computed at the same time as the bodies motions, in a coupled manner. One possible simplification is to make a quasi-static assumption and evaluate the lift/drag force on the foil based solely on the bodies instantaneous velocity and angle of attack. Even with the quasi-static assumption, there are still significant computational requirements because the foil operates at both a wide range or Reynolds number and a wide range of angles of attack (-180 degrees - > 180degrees). Nevertheless, this approach is very attractive because it would allow the foil flow solution to be computed independently of bodies motion and allow a particular foils computational results to be re-used for more than one simulation that varies parameters unrelated to the foil such as wave-height or buoy size. However, the validity of the quasi-static assumption is currently unknown and needs to be validated before simulation results using this approach can be considered substantial.

Figure 1 illustrates one frame of a simulation that has been completed utilizing the quasi-static assumption and lift/drag coefficient curves for a single Reynolds number. The blue arrow in the lower right shows the flow angle of attack and the green arrow shows the resultant lift/drag vector computed from the lift and drag coefficients computed for a NACA 0015 foil at all angles of attack and at Re = $1X10^6$, see Figure 2.

An animation of the simulation can be seen at the URL cited in the caption for Figure 1. In the animation, the lift/drag force vector is colored green when it is point to the right and is providing net propulsion, and red when it is causing net drag. Not shown in the animation is the tether between the buoy and the submerged foil, it is there in the simulation and is driving the foil motions. The waves are not insignificant this simulation (significant wave height = 4m) and cause the non-streamlined buoy to advance into the waves and a simulated 15kt wind. The lift behavior of the foil and its effectiveness can be seen in the small percentage of time the net force vector is red, even during the times when the flow is predominately from the right relative to the foil. Figure 3 shows the position history of the foil for a longer time than is shown in the simulation, the average rate of advance is about knots.

These results are encouraging but the validity of the quasistatic assumption is in question, and the lift and drag coefficients of the foil will most certainly change across the range of Reynolds number the foil encounters throughout the solution. Figure 4 illustrates the Reynolds number of the instantaneous flow over the foil based upon chord length at all force evaluations throughout the 600 second simulation. These results provide an estimate of the range of Reynolds numbers the CFD computations will be required at. From a design point of view, a figure of merit for the pitch control algorithm or mechanism will be the percentage of the time the foil is operating at an optimum angle of attack, i.e. 12 degrees or so.

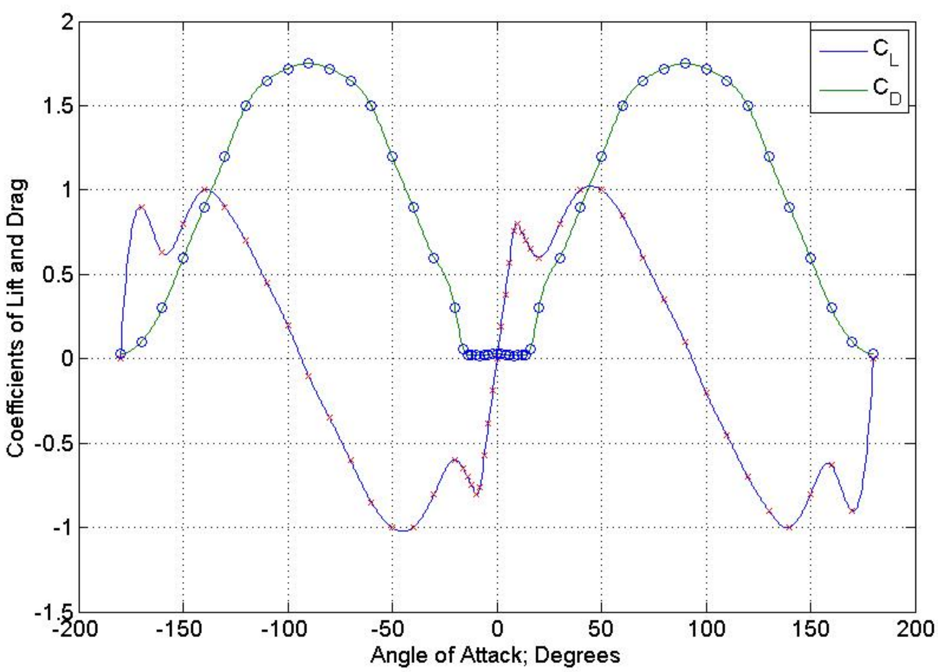

Figure 2: Lift and drag coefficients for a NACA 0015 foil at all angles of attack, Re = $1X10^6$. [data from Sheldahl, Klimas, Sandia NL, 1981]

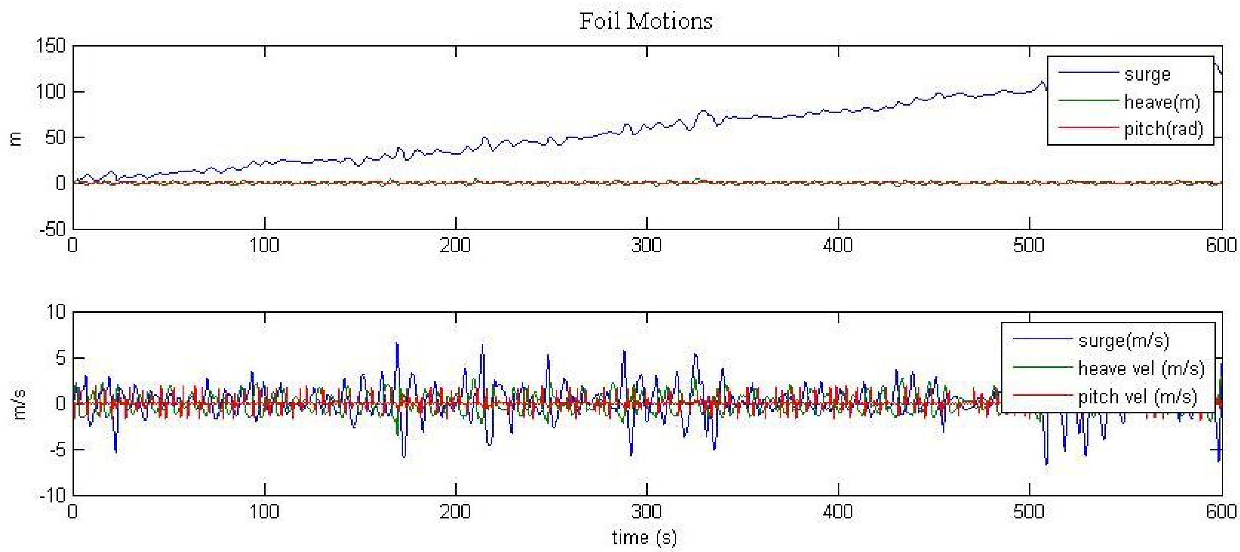

Figure 3: Foil position and velocity versus time.

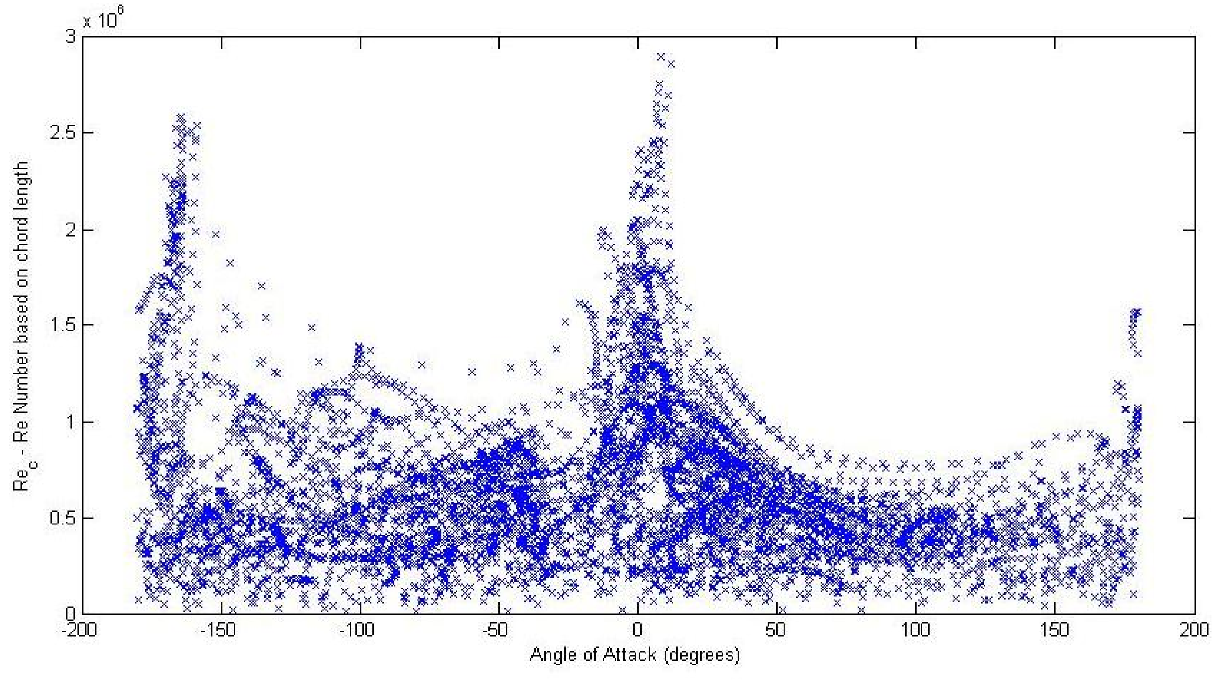

Figure 4: Reynolds number versus angle of attack for all force evaluations during 600 second simulation.

Suggested Work Plan

The work required falls into two phases, one to determine what assumptions are reasonable in the solution technique, and one to compute results for a selection of cases that will help us with the design component of the project. The first phase will require performing sufficient computations (and possible testing) to determine if the quasi-static assumption is valid and to determine what granularity of Reynolds number and angle of attack is required. To do this we could compute a set of results utilizing the quasi-static assumption for a specific foil size and shape, a specific buoy size and shape, and specific sea conditions. Subsequent to that we could compute a fully coupled transient solution in which solvers run by Airflow Sciences are coupled at each timestep (actually four times per timestep due to the 4^{th} order nature of the solver employed). Depending on the results from these simulations we may consider performing some physical testing to validate the model results. In this case we'd probably initially consider a carefully controlled experiment that focused on the transient flapping foil problem in controlled conditions, as opposed to a broader test that would deploy a device and measure gross parameters such as rate of advance in various sea states.

Presuming we are able to develop a computational approach we are confident in, the second phase of the work will involve running a variety of model configurations in a variety of sea conditions.

Practicalities

For the quasi-static approach, Airflow sciences will be producing lift/drag curves for a particular foil (2d or 3d as deemed appropriate) at a range of Reynolds number. In this case the foil characteristics and expected Reynolds number range will be supplied by MBARI and the results can be returned in any convenient format.

In the transient case, a mechanism for communication between the MBARI solver and Airflow Sciences codes will need to be worked out. The MBARI solver is written in Matlab and can be set up to employ just about any type of communication scheme, ranging from a simple flat file handshaking that would require the two codes to run on the same computer (or at least have access to the same disk), to a more sophisticated messaging scheme based upon internet sockets or similar. This would allow each code to be run at the home institution if desired.