Proceedings of the 1999 IEEE
International Symposium on Computer Aided Control System Design
Kohala Coast-Island of Hawai'i, Hawai'i, USA • August 22-27, 1999

A Computer Algebra Approach to Undersea Vehicle Dynamics

H. G. Kwatny
Department of Mechanical Engineering & Mechanics
Drexel University
Philadelphia, PA 19104

E. Salter
Techno-Sciences, Inc.
10001 Derekwood Lane, Suite 204
Lanham, MD 20706

E. S. Ammeen
Carderock Division
NSWC
9500 MacArthur Blvd.
West Bethesda, MD 20817-5700

G. L. Blankenship
Department of Electrical & Computer Engineering
University of Maryland
College Park, MD 20742

Abstract: Symbolic computing can facilitate the application of modern nonlinear system analysis and design methods to engineering problems. Reasonably complex models can be efficiently assembled and manipulated. In this paper we illustrate the symbolic construction and manipulation of a model of an undersea vehicle. While the system considered here is within the realm of hand assembly, doing so is tedious and error prone. On the other hand, it is trivial with the symbolic computing tools described here. More complex, multibody configurations of undersea vehicles and robots can be dealt with using these techniques.

1 Introduction

The dynamical behavior of undersea vehicles has been a subject of considerable interest for many years. Modern methods of nonlinear dynamics provide new analytical tools that could yield significant new understanding of these complex dynamics and their control. In a series of papers [1-3] Papoulias and his students have applied local bifurcation analysis to study the stability of undersea vehicles. In another line of inquiry, Leonard has applied Lie algebraic methods to study the global dynamics and nonlinear control of submerged rigid bodies [4-6]. Fossen and his colleagues [7-9] have considered a variety of nonlinear control problems associated with surface and undersea vehicles.

In order to realize the full benefits of these techniques for full-scale operational vehicles, efficient tools are required to assemble and manipulate detailed and accurate models. Symbolic computing has matured to the point that it is suitable for such applications. In recent studies we investigated dynamical properties of undersea vehicles using existing computer algebra tools for multibody modeling and nonlinear control [10-13]. Our results include:

- Computer assembly of a symbolic model,
- Computer generation of a simulation model,
- Computer derivation of simple, computable approximations to certain singular drag integrals,
- A stability analysis of the open loop dynamics, and
- A controllability analysis.

The strategy is to develop a 'first principles' mathematical model using computer algebra methods from which a simulation model is constructed. Thus, a consistent pair of (analytical and computational) models is available for further analysis as appropriate. In the present investigation, all of the symbolic constructions are carried out using the symbolic computing software package *Mathematica* supplemented with modeling and control tools described in [10-13]. We call this set of tools *ProPac*. The simulation model is a (computationally optimized) C source-code implementation of the mathematical model that compiles as a Simulink S-function defining a vehicle module in the Simulink block-diagramming environment. Models assembled in this way are easily modified to add complexity by including additional components or adding neglected physical effects.

The symbolic model can be manipulated to derive various reduced complexity models including linear models and/or it can be used for diverse analysis and control design processes. The computational model can be used to simulate event or disturbance responses, for model validation, and to evaluate proposed control strategies.

In this investigation the specific vehicle of interest is the Naval Postgraduate School Autonomous Underwater Vehicle II (NPSAUVII) which has been scaled to be more representative of a military submarine. The NPSAUVII is convenient because it has been well documented and frequently used in dynamics and control studies. Our main source of data has been the book [7]. All of our notation is consistent with that book (specifically, Appendix E.2.3). We emphasize that the significance of the approach described here is its potential for application to more complex multibody configurations of undersea vehicles and robots.

2 Model Assembly

The dynamical equations are derived in Poincaré's form of Lagrange's equations:

Kinematics: $$\dot{q} = V(q)p \qquad (1)$$

Dynamics: $$M(q)\dot{p} + C(q,p)p + F(q,p,u) = 0 \qquad (2)$$

where q is a vector of configuration coordinates, p is a vector of quasi–velocities and u is a vector of exogenous inputs. A discussion of the methods we use can be found in [12]. General information about Poincaré's equations can be found in [14, 15]. Equations of this type are sometimes called Lagrange's equations in quasi-coordinates [16]. Such a formulation for ship dynamics is used by Fossen [7]. In the following paragraphs we describe development of the submarine model.

2.1 Kinematics

The kinematic model is very simple. It consists of a single 6 degree of freedom joint between the inertial frame and a body fixed frame with origin at the center of mass. The *Mathematica* code that defines the joint and computes all of the required kinematic parameters is:

```
In[3]:= r1 = {6};
    H1 = IdentityMatrix[6];
    q1 = {ϕ, θ, ψ, x, y, z}; p1 = {p, q, r, u, v, w};
    JointList = {{r1, H1, q1, p1}};
    {JV, JX, JH} = Joints[JointList];
```

Here, the output of the *ProPac*[1] function `Joints` include `JV`, the matrix $V(q)$, required in equation (1), and `JX`, the submarine configuration matrix $X(q)$. Notice that the configuration coordinates $q_1 = [\phi \quad \theta \quad \psi \quad x \quad y \quad z]$. Because of our joint definition these are the body Euler angles (3-2-1 convention) and the body center of mass coordinates in the inertial frame. The quasi-velocities are $p_1 = [p \quad q \quad r \quad u \quad v \quad w]$. Here p, q, r are the angular velocity components along the body-fixed axes and u, v, w are the translational velocity components in body coordinates.

[1] More information about *ProPac* and a *Mathematica* notebook containing the computations described herein may be found at www.technosci.com.

2.2 Dynamics

ProPac includes modeling tools for multibody mechanical systems composed of both rigid and flexible bodies. The fluid loading on a rigid submerged vehicle can alter the inertia matrix in a general way, possibly coupling the translational and rotational parts of the matrix. Even if this is not the case, the translational inertias in the *x*-*y*-*z* directions will almost surely be different from each other. That is not the case in a free rigid body. The dynamics tools in *ProPac* can deal with these cases by using the flexible body data structure for defining the body.

In the following, we build the model assuming that generalized forces act on the system: *K*, *M*, *NN* are moments acting about the body *x*, *y*, *z* axes, repectively, and *X*, *Y*, *Z* are forces acting in the body *x*, *y*, *z* directions. These generalized forces arise from propulsion and nonconservative fluid dynamics effects including lift and drag forces acting on the body and its control surfaces. They are described in more detail in subsequent paragraphs.

```
In[5]:= VehicleMass = m;
  CenterOfMass = Transpose[{{xg, yg, zg}}];
  Out1 = Transpose[{{0, 0, 0, xb, yb, zb}}];
  T1 = BlockMatrix[
    {{IdentityMatrix[3] + L, ZeroMatrix[3]}, {ZeroMatrix[3], IdentityMatrix[3]}}];
  FluidInertia = T1.{
    {-Kpdot, 0, -Npdot, 0, -Kvdot, 0},
    {0, -Mqdot, 0, 0, 0, -Mwdot},
    {-Npdot, 0, -Nrdot, 0, -Nvdot, 0},
    {0, 0, 0, -Xudot, 0, 0},
    {-Kvdot, 0, -Nvdot, 0, -Yvdot, 0},
    {0, -Mwdot, 0, 0, 0, -Zwdot}}.T1*ρ*L^3/2;
  Inertiacg = {{Ix, -Ixy, -Ixz}, {-Ixy, Iy, -Iyz}, {-Ixz, -Iyz, Iz}};
  Inertia = BlockMatrix[{{Inertiacg, m*XToXTilde[{xg, yg, zg}]},
     {-m*XToXTilde[{xg, yg, zg}], m*IdentityMatrix[3]}}] + FluidInertia;
In[6]:= Damping = 0*IdentityMatrix[6];
  Stiffness = 0*IdentityMatrix[6];
  Body1 = {CenterOfMass, {{2, Out1}}, m, {Inertia, Damping, Stiffness}, {}, {}};
  BodyList = {Body1};
In[7]:= TreeList = {{{1, 1}}};
  CenterOfBouyancy = EndEffector[2, TreeList, BodyList, JX][[3, 4]];
  PE = g*ρ*V*CenterOfBouyancy;
  Q = {K, M, NN, X, Y, Z};
  {JV, JX, JH, MM, Cp, Fp, pp, qq} =
   CreateModel[JointList, BodyList, TreeList, -g, PE, Q, JV, JX, JH];
```

The outputs of the *ProPac* function CreateModel include the MM, Cp, and Fp. These are, respectively the inertia matrix, $M(q)$, the matrix, $C(q,p)$, and the vector $F(q,p,u)$. These are the parameters that make up Equation (2). The output of these calculations are given in Appendix 1.

2.3 External Forces

The model is completed by defining the generalized force components, the moments: K, M, NN, and the forces: X, Y, Z. The gravitational and buoyancy effects have already been accounted for. What remain are due to propulsion and fluid dissipation. The key elements are propulsion, control surface lift and drag forces, body skin friction and body form drag. For convenience we divide Q into four parts

$$Q = Q_{propulsion} + Q_{control\ surfaces} + Q_{body\ friction} + Q_{body\ drag}$$

These force components are established empirically. Except for the last term we follow the formulation of Fossen [7] with minor changes. However, we treat body drag somewhat differently because we want an explicit analytical model, not just a simulation model. Consequently, we give a brief discussion of the drag computations but not the other components of the external force model. A summary of the external forces as assembled is included in the Appendix.

Various models for undersea vehicles including that of NPSAUVII require the evaluation of certain integrals to obtain drag forces in the heave, sway, pitch and yaw equations. These drag integrals can be integrated to provide explicit functions characterizing each drag force in terms of the four variables: heave velocity, sway velocity, pitch rate and yaw rate. However, the expressions are singular at the origin which is the nominal operating condition. Consequently, some form of approximation is used for computational purposes.

Let v, w, p, r represent sway, heave, pitch and yaw velocities, and D_{heave}, D_{sway}, D_{pitch}, and D_{yaw} represent the respective drag forces. The drag integrals are defined by

$$\begin{aligned}
D_{heave} &= \int_{x_{tail}}^{x_{nose}} K(x;v,w,q,r)(w-xq)dx \\
D_{sway} &= \int_{x_{tail}}^{x_{nose}} K(x;v,w,q,r)(v+xr)dx \\
D_{pitch} &= \int_{x_{tail}}^{x_{nose}} K(x;v,w,q,r)(w-xq)xdx \\
D_{yaw} &= \int_{x_{tail}}^{x_{nose}} K(x;v,w,q,r)(v+xr)xdx
\end{aligned} \tag{3}$$

where the kernel is

$$K(x;v,w,q,r) = \frac{c_{dy}h(v+xr)^2 + c_{dz}b(w-xq)^2}{\sqrt{(v+xr)^2+(w-xq)^2}} \tag{4}$$

These integrals can be evaluated to yield explicit, although complex, formulas. See Appendix 2. However, the formulas are singular at the origin ($v=0, w=0, q=0, r=0$). Of course, it would be preferable to have computable and accurate formulas around the origin, particularly because this is the usual operating region. The singularity is not essential. In fact, the integrals evaluate to zero at the origin, which is easily seen by substituting zero for the velocities before integration. Moreover, the limiting values of the integral formulas are zero which suggests that the integrals are continuous. On the other hand, they are not continuously differentiable which rules out a Taylor series representation.

We derive simple, nonsmooth formulas for the drag integrals. They provide approximations that are computable and accurate around the origin. It has already been noted that a Taylor series does not exist at the origin. However, if one of the four velocities is assumed not to be zero, a Taylor series does exist in the other three around zero. Consequently, we follow a simple strategy: to each integral we associate its natural primary velocity: heave- w, sway- v, pitch- q, yaw- r. Then we develop a Taylor approximation in the other three variables and simplify. Using computer algebra calculations we obtain (with $x_{tail} = -L, x_{nose} = L$):

$$\begin{aligned}
D_{heave} &= -\frac{1}{3}L\begin{pmatrix} 2c_{dy}h\left(L^2r^2+3v^2\right) \\ +bc_{dz}\left(L^2\left(2q^2-r^2\right)-3v^2+6w^2\right)\end{pmatrix}\operatorname{sgn}(w) \\
D_{sway} &= \frac{1}{3}L\begin{pmatrix} -2bc_{dz}\left(L^2q^2+3w^2\right) \\ +c_{dy}h\left(L^2\left(q^2-2r^2\right)-6v^2+3w^2\right)\end{pmatrix}\operatorname{sgn}(v) \\
D_{pitch} &= -\frac{2}{3}L^3\left(-2c_{dy}hrv + bc_{dz}(rv+2qw)\right)\operatorname{sgn}(q) \\
D_{yaw} &= -\frac{2}{3}L^3\left(-2bc_{dz}qw + c_{dy}h(2rv+qw)\right)\operatorname{sgn}(r)
\end{aligned}$$

These formulas are derived using 2[nd] order approximations for heave and sway and 3[rd] order for pitch and yaw (in the secondary variables). They are simple, computable and accurate around the origin, and they capture the basic physics reflected in the formulation leading to the drag integrals.

3 Stability of Equilibria

In addition to simulation studies, we have performed (local) stability analysis of all of these models. This has been done at the symbolic level, where we have derived parameter dependent linear models at specified equilibria. This has allowed us to corroborate the simulation results and to identify specific parameters that affect stability. It also underscores the value of being able to work efficiently with the symbolic equations.

Suppose, that (u_0, p_0, q_0) is an equilibrium point about which we compute the linearization:

$$
\begin{aligned}
\delta\dot{q} &= V_0 \delta p \\
M_0 \delta\dot{p} + C_0 \delta p + K_0 \delta q &= 0
\end{aligned}
\tag{5}
$$

We performed these calculations for the equilibrium case of straight and level (no roll) horizontal motion at constant speed u_0 corresponding to engine speed n_0 and with all control surface deflections set to zero. This results in $M_0^T = M_0 > 0, K_0^T = K_0 \geq 0$ and both matrices are independent of the speed u_0. In addition, we find $C_0 = C_{00}|u_0|$, where C_{00} is independent of u_0 but is not symmetric. These calculations are included in the accompanying notebook. We can make use of a generalization of Lagrange's classic theorem [17] that states that under the asserted conditions and if q_0 corresponds to an isolated minimum of the potential energy function[2], then the equilibrium point is stable if and only if symmetric$[C_0] \geq 0$. That is, the symmetric part of C_0 (the dissipation part as opposed to the gyroscopic part) must be nonnegative. However, we find that symmetric$[C_{00}]$ has a single negative eigenvalue so that the test fails for any $u_0 \neq 0$.

```
In[80]:= CC = (Jacobian[Cp.pp, pp] /. {p -> 0, q -> 0, r -> 0, w -> 0, v -> 0}) /. {u -> Abs[u]};
         CCs = (CC + Transpose[CC]) / 2;
         BB = Simplify[(UnitStep2Sign[
                 Jacobian[Fp /. {δr -> 0, δs -> 0, δb -> 0, δbs -> 0, δbp -> 0}, {p, q, r, u, v, w}] /.
                  {p -> 0, q -> 0, r -> 0, v -> 0, w -> 0}] /. EqRule[[1]]) /.
             Sign[n] -> Sign[u]];
         (CCs + BB) // TraditionalForm
```

Out[80]//TraditionalForm=

$$
\begin{pmatrix}
3.8451\times10^{8}|u| & 0 & 2.72652\times10^{7}|u| & 0 & -1.18506\times10^{6}|u| & 0 \\
0 & 2.37697\times10^{9}|u| & 0 & 0 & 0 & 4.40551\times10^{6}|u| \\
2.72652\times10^{7}|u| & 0 & 5.17341\times10^{8}|u| & 0 & 1.65815\times10^{7}|u| & 0 \\
0 & 0 & 0 & 32190.9|u| & 0 & 0 \\
-1.18506\times10^{6}|u| & 0 & 1.65815\times10^{7}|u| & 0 & 418064.|u| & 0 \\
0 & 4.40551\times10^{6}|u| & 0 & 0 & 0 & 1.25419\times10^{6}|u|
\end{pmatrix}
$$

```
In[81]:= Eigenvalues[(CCs + BB) / Abs[u]]
Out[81]= {2.37698×10^9, 5.23212×10^8, 3.79181×10^8, 1.24602×10^6, -124331., 32190.9}
```

4 Simulations

Simulation experiments were conducted with three variations of the scaled NPSAUVII model: a full 6 degree of freedom model, a 3 degree of freedom model/motion restricted to the vertical plane, a 3 degree of freedom model/motion restricted to the horizontal plane. Building a simulation model as an S-function MEX file for use in SIMULINK is very simple. Here is the *Mathematica* code:

```
In[39]:= Inputs = {deltar, deltas, deltab, deltabp, deltabs, n};
         Outputs = {p, q, r, u, v, w, phi, theta, psi, x, y, z};
         MEXFilename = "NPSAUVII.c";
         PassedParams = {"X0"};
         PassedParamsDimensions = {{Length[p1] + Length[q1], 1}};
         CreateModelMEX[pp, qq, Inputs, Outputs, PassedParams,
             PassedParamsDimensions, JV, Cp,
             Fp, MM, MEXFilename];
```

In the following paragraphs we will illustrate just of few of many simulations conducted. All simulations were run open loop with various settings for engine speed, rudder and stern planes. Computations appear to be very efficient and fast.

The simulations clearly show a divergence instability in yaw. Analysis of the model confirms that this is indeed the case and the instability is not a computational anomaly.

We will describe some results using the six degree of freedom model. In this case, all surfaces are set to zero except the stern plane which has a deflection of 1 deg. Propeller speed is 1000 rpm (note that we have not scaled the propulsion equations so that this produces reasonable speeds given the increased forces of the scaled model).

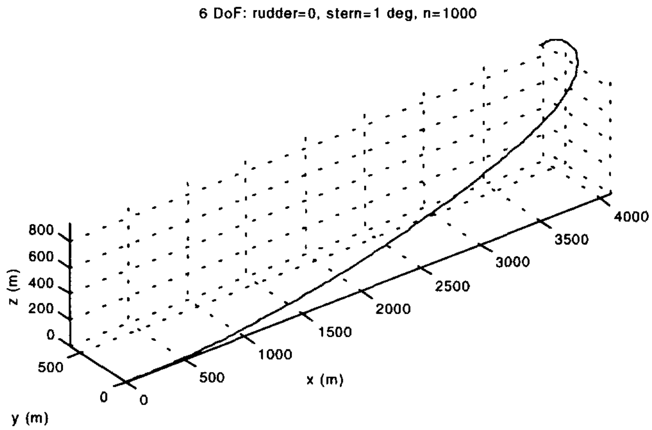

Figure 1 This xyz plot of the position of the vehicle clearly shows the yaw divergence instability. It is helpful to remember that the vehicle is about 100m in length.

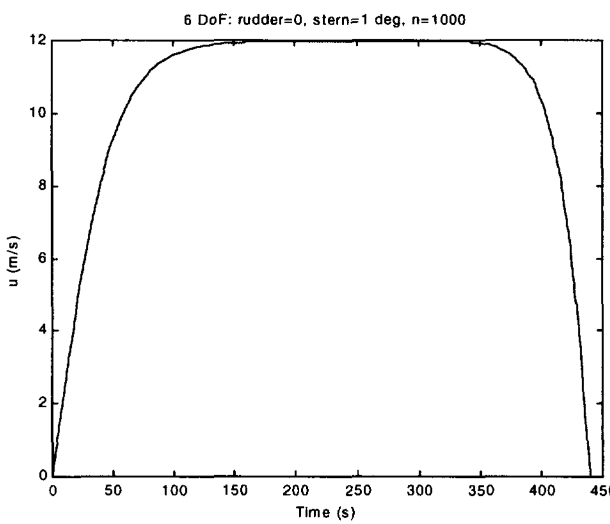

Figure 2 The forward velocity increases to its steady state value and the dramatically drops as the vehicle spin develops.

5 Conclusions

Although we are primarily interested in the complete, nonlinear six degree of freedom (dof), system we found it convenient to investigate the behavior of two three dof models as well. We have built the following models: a 6 degree of freedom model, a 3 degree of freedom model in the vertical plane, a 3 degree of freedom model in the horizontal plane. It is common to use linearized versions of the vertical plane model for pitch and depth control system design and the horizontal plane model for heading control system design.

We performed limited computational experiments with all three models. These consisted of various initial conditions with nonzero constant propeller speed, and several constant values for rudder, stern plane and bow planes. Initial experiments with the 6 dof model indicated an instability in yaw dynamics. This prompted

[2] There is no essential contradiction in the requirement of an isolated minimum of the potential energy and an indefinite K_0, even if the potential energy is quadratic, because q can be of smaller dimension than p in this setup.

us to reexamine the data and to experiment with the three degree of freedom models. While we found some inconsistencies in the data, appropriate changes did not alter the basic qualitative behavior. The 3 dof models behaved similarly to the 6 dof model. Resorting to the original unscaled data also did not alter the qualitative behavior. We also found that we could generate a pitch instability at slow speed by placing the center of buoyancy at the center of mass.

In addition to simulation studies, we performed (local) stability analysis of all of these models. This has been done at the symbolic level, where we have derived parameter dependent linear models at specified equilibria. This has allowed us to corroborate the simulation results and to identify specific parameters that affect stability. It also underscores the value of being able to work efficiently with the symbolic equations. The fact that instabilities of this type may exist is not surprising. Underwater vehicle stability has been studied in six degrees of freedom in recent years, specifically in [3] and also [6, 18], and clearly show how such instabilities can arise.

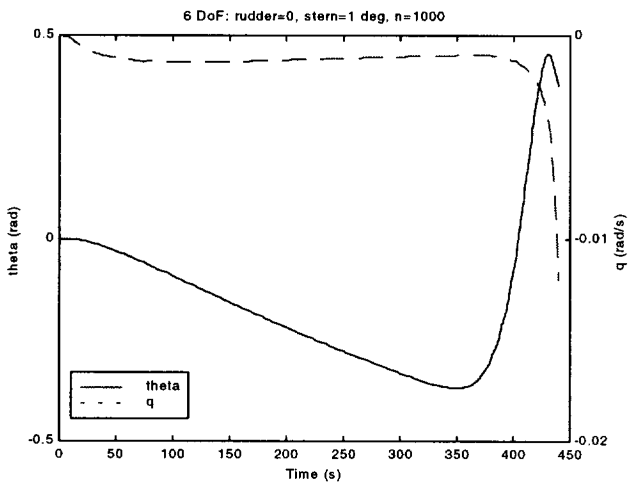

Figure 3 Pitch behavior also shows the eventual effects of spin as it develops. See the three degree of freedom vertical plane model below for comparison.

Acknowledgement: Research supported by the U. S. Navy, Contract Number N00167-97-M-0809

6 References

1. Papoulias, F.A., C.A. Bateman, and S. Ornek, *Dynamic Loss of Stability in Depth Control of Submersible Vehciles.* Applied Ocean Research, 1995. **17**: p. 205-216.

2. Papoulias, F.A. and H.A. Papadimitriou, *Nonlinear Studies of Dynamic Stability of Submarines in the Dive Plane.* Journal of Ship Research, 1995. **39**(4): p. 347-356.

3. Papoulias, F.A. and I. Aydin, *Out-of-Plane Solutions and Bifurcations of Submersibles in Free Positive Bouyancy Ascent.* Journal of Ship Research, 1994. **38**(4): p. 259-271.

4. Leonard, N.E. and P.S. Krisnaprasad, *Motion Control of Drift-Free, Left-Invarant Systems on Lie Groups.* IEEE Transactions on Automatic Control, 1995. **40**(9): p. 1539-1554.

5. Leonard, N.E., *Control Synthesis and Adaptation for an Underactuated Autonomous Underwater Vehicle.* IEEE Journal of Ocean Engineering, 1995. **20**(3): p. 211-220.

6. Leonard, N.E., *Stability of a Bottom-Heavy Underwater Vehicle.* Automatica, 1997. **33**(3): p. 331-346.

7. Fossen, T.I., *Guidance and Control of Ocean Vehicles.* 1994, New York: John Wiley and Sons.

8. Fossen, T.I. and A. Grovlen, *Nonlinear Output Feedback Control of Dynamically Positioned Ships Using Vectorial Observer Backstepping.* IEEE Transactions on Control System technology, 1998. **TCST-6**(1): p. 121-128.

9. Vik, B. and T.I. Fossen, *Semi-Global Exponential Output Feedback Control of Ships.* IEEE Transactions on Control System Technology, 1997. **TSCT-5**(3): p. 360-370.

10. Kwatny, H.G. and C. LaVigna, *TSi Dynamics User's Guide*, . 1994, Techno–Sciences, Inc.: Lanham, MD.

11. Kwatny, H.G. and G.L. Blankenship, *TSi Controls User's Guide*, . 1994, Techno-Sciences, Inc.: Lanham, MD.

12. Kwatny, H.G. and G.L. Blankenship, *Symbolic Construction of Models for Multibody Dynamics.* IEEE Transactions on Robotics and Automation, 1995. **11**(2): p. 271-281.

13. Blankenship, G.L., *et al.*, *Integrated tools for Modeling and Design of Controlled Nonlinear Systems.* IEEE Control Systems, 1995. **15**(2): p. 65-79.

14. Arnold, V.I., V.V. Kozlov, and A.I. Neishtadt, *Mathematical Aspects of Classical and Celestial Mechanics.* Encyclopedia of Mathematical Sciences, ed. V.I. Arnold. Vol. 3. 1988, Heidelberg: Springer–Verlag.

15. Chetaev, N.G., *Theoretical Mechanics.* 1989, New York: Springer–Verlag.

16. Meirovitch, L., *Methods of Analytical Dynamics.* 1970, New York: McGraw–Hill, Inc.

17. Gantmacher, F., *Lectures in Analytical Mechanics.* English Translation ed. 1975, Moscow: Mir.

18. Holmes, P., J. Jenkins, and N.E. Leonard, *Dynamics of the Kirchoff Equations I: Coincident Centers of Gravity and Bouyancy*, . 1997, Mechanical and Aerospace Engineering, Princeton University: Princeton.

Appendix 1: Sample Results of Model Assembly

Kinematics, $V(q)$

$$\begin{pmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 & 0 & 0 \\ 0 & \sec(\theta)\sin(\phi) & \cos(\phi)\sec(\theta) & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(\theta)\cos(\psi) & \cos(\psi)\sin(\theta)\sin(\phi)-\cos(\phi)\sin(\psi) & \cos(\phi)\cos(\psi)\sin(\theta)+\sin(\phi)\sin(\psi) \\ 0 & 0 & 0 & \cos(\theta)\sin(\psi) & \cos(\phi)\cos(\psi)+\sin(\theta)\sin(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\sin(\psi)-\cos(\psi)\sin(\phi) \\ 0 & 0 & 0 & -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{pmatrix}$$

Inertia Matrix, $M(q)$

$$\begin{pmatrix}
\text{Ix}-\frac{1}{2}\,\text{Kpdot}\,L^5\rho & -\text{Ixy} & -\frac{1}{2}\,\text{Npdot}\,\rho L^5-\text{Ixz} & 0 & -\frac{1}{2}\,\text{Kvdot}\,\rho L^4-m\,\text{zg} & m\,\text{yg} \\
-\text{Ixy} & \text{Iy}-\frac{1}{2}L^5\,\text{Mqdot}\,\rho & -\text{Iyz} & m\,\text{zg} & 0 & -\frac{1}{2}\,\text{Mwdot}\,\rho L^4-m\,\text{xg} \\
-\frac{1}{2}\,\text{Npdot}\,\rho L^5-\text{Ixz} & -\text{Iyz} & \text{Iz}-\frac{1}{2}L^5\,\text{Nrdot}\,\rho & -m\,\text{yg} & m\,\text{xg}-\frac{1}{2}L^4\,\text{Nvdot}\,\rho & 0 \\
0 & m\,\text{zg} & -m\,\text{yg} & m-\frac{1}{2}L^3\,\text{Xudot}\,\rho & 0 & 0 \\
-\frac{1}{2}\,\text{Kvdot}\,\rho L^4-m\,\text{zg} & 0 & m\,\text{xg}-\frac{1}{2}L^4\,\text{Nvdot}\,\rho & 0 & m-\frac{1}{2}L^3\,\text{Yvdot}\,\rho & 0 \\
m\,\text{yg} & -\frac{1}{2}\,\text{Mwdot}\,\rho L^4-m\,\text{xg} & 0 & 0 & 0 & m-\frac{1}{2}L^3\,\text{Zwdot}\,\rho
\end{pmatrix}$$

Dynamics, $C(q,p)p$

$$\begin{pmatrix}
-m\,q\,\text{yg}-m\,r\,\text{zg} \\
\frac{1}{2}\,\text{Mwdot}\,q\,\rho L^4-\frac{1}{2}\,w\,\text{Xudot}\,\rho L^3+\frac{1}{2}\,w\,\text{Zwdot}\,\rho L^3+m\,q\,\text{xg} \\
-\frac{1}{2}\,\text{Kvdot}\,p\,\rho L^4-\frac{1}{2}\,\text{Nvdot}\,r\,\rho L^4+\frac{1}{2}\,v\,\text{Xudot}\,\rho L^3-\frac{1}{2}\,v\,\text{Yvdot}\,\rho L^3+m\,r\,\text{xg} \\
0 \\
m\,r-\frac{1}{2}L^3\,r\,\text{Xudot}\,\rho \\
\frac{1}{2}L^3\,q\,\text{Xudot}\,\rho-m\,q
\end{pmatrix}$$

Dynamics, $F(q,p,u)$

$$\begin{pmatrix}
-K-g\cos(\theta)\left((m\,\text{yg}-V\,\text{yb}\,\rho)\cos(\phi)+(V\,\text{zb}\,\rho-m\,\text{zg})\sin(\phi)\right) \\
\frac{1}{2}\left(g\,(m\,\text{xg}-V\,\text{xb}\,\rho)\cos(\theta-\phi)+g\,(m\,\text{xg}-V\,\text{xb}\,\rho)\cos(\theta+\phi)-2\,(M-g\,m\,\text{zg}\sin(\theta)+g\,V\,\text{zb}\,\rho\sin(\theta))\right) \\
\frac{1}{2}\left(-2\,\text{NN}-2g\,(m\,\text{yg}-V\,\text{yb}\,\rho)\sin(\theta)+g\,(m\,\text{xg}-V\,\text{xb}\,\rho)\sin(\theta-\phi)-g\,m\,\text{xg}\sin(\theta+\phi)+g\,V\,\text{xb}\,\rho\sin(\theta+\phi)\right) \\
g\,(m-V\rho)\sin(\theta)-X \\
-Y-g\,(m-V\rho)\cos(\theta)\sin(\phi) \\
-Z-g\,(m-V\rho)\cos(\theta)\cos(\phi)
\end{pmatrix}$$

External Forces, $Q_{propulsion}+Q_{control\ surfaces}=Q_1+Q_2 e$, $e=\left(\text{sgn}(n)\sqrt{c_t+1}\big/\text{sgn}(u)\sqrt{c_{t1}+1}\right)-1$

Q_1:

$$\begin{pmatrix}
0 \\
\frac{1}{2}L^3\,u\,(\text{Mdb2}\,(\delta\text{bp}+\delta\text{bs})+\text{Mds}\,\delta\text{s})\,\rho\,|u| \\
\frac{1}{2}L^3\,\text{Ndr}\,u\,\delta\text{r}\,\rho\,|u| \\
\frac{1}{2}L^2\,u\,\rho\,(w\,\text{Xwdb2}\,\delta\text{bp}+w\,\text{Xwdb2}\,\delta\text{bs}+v\,\text{Xvdr}\,\delta\text{r}+w\,\text{Xwds}\,\delta\text{s}+L\,(q\,\text{Xqdb2}\,\delta\text{bp}+q\,\text{Xqdb2}\,\delta\text{bs}+r\,\text{Xrdr}\,\delta\text{r}+q\,\text{Xqds}\,\delta\text{s})+(\text{Xdbdb2}\,\delta\text{b}^2+\text{Xdrdr}\,\delta\text{r}^2+\text{Xdsds}\,\delta\text{s}^2+\text{Xprop})\,|u|) \\
\frac{1}{2}L^2\,u\,\text{Ydr}\,\delta\text{r}\,\rho\,|u| \\
\frac{1}{2}L^2\,u\,(\text{Zdb2}\,(\delta\text{bp}+\delta\text{bs})+\text{Zds}\,\delta\text{s})\,\rho\,|u|
\end{pmatrix}$$

Q_2:

$$\begin{pmatrix}
\frac{1}{2}\,\text{Kpn}\,L^4\,p\,\rho\,|u| \\
\frac{1}{2}\,\text{Mqn}\,q\,\rho\,|u|\,L^4+\frac{1}{2}\,\text{Mwn}\,w\,\rho\,|u|\,L^3+\frac{1}{2}\,\text{Mdsn}\,u\,\delta\text{s}\,\rho\,|u|\,L^3 \\
0 \\
\frac{1}{2}\,q\,u\,\text{Xqdsn}\,\delta\text{s}\,\rho L^3+\frac{1}{2}\,u\,w\,\text{Xwdsn}\,\delta\text{s}\,\rho L^2+\frac{1}{2}\,u\,\text{Xdsdsn}\,\delta\text{s}^2\,\rho\,|u|\,L^2 \\
0 \\
\frac{1}{2}\,q\,\text{Zqn}\,\rho\,|u|\,L^3+\frac{1}{2}\,w\,\text{Zwn}\,\rho\,|u|\,L^2+\frac{1}{2}\,u\,\text{Zdsn}\,\delta\text{s}\,\rho\,|u|\,L^2
\end{pmatrix}$$

$Q_{body\ friction}$

$$\begin{pmatrix}
-\frac{1}{2}\,\text{Kpp}\,\rho\,|u|\,L^4-\frac{1}{2}\,\text{Krr}\,\rho\,|u|\,L^4-\frac{1}{2}\,\text{Kvv}\,\rho\,|u|\,L^3 \\
\frac{1}{2}L^4\,(-\text{Muq}-\text{Mwdot})\,q\,\rho\,|u|-\frac{1}{2}L^3\,\text{Mww}\,\rho\,|u| \\
-\frac{1}{2}\,\text{Npp}\,\rho\,|u|\,L^4-\frac{1}{2}\,\text{Nrr}\,\rho\,|u|\,L^4-\frac{1}{2}\,\text{Nvv}\,\rho\,|u|\,L^3 \\
0 \\
-\frac{1}{2}\,p\,\text{Yp}\,\rho\,|u|\,L^3-\frac{1}{2}\,r\,\text{Yr}\,\rho\,|u|\,L^3-\frac{1}{2}\,v\,\text{Yv}\,\rho\,|u|\,L^2 \\
-\frac{1}{2}\,q\,\text{Zq}\,\rho\,|u|\,L^3-\frac{1}{2}\,w\,\text{Zw}\,\rho\,|u|\,L^2
\end{pmatrix}$$

$Q_{body\ drag}$

$$\begin{pmatrix}
0 \\
-\frac{2}{3}L^3\,(b\,\text{Cdz}\,(r\,v+2\,q\,w)-2\,\text{Cdy}\,h\,r\,v)\,\text{sgn}(q) \\
-\frac{2}{3}L^3\,(\text{Cdy}\,h\,(2\,r\,v+q\,w)-2\,b\,\text{Cdz}\,q\,w)\,\text{sgn}(r) \\
0 \\
\frac{1}{3}L\,(\text{Cdy}\,h\,((q^2-2r^2)L^2-6\,v^2+3\,w^2)-2\,b\,\text{Cdz}\,(L^2q^2+3\,w^2))\,\text{sgn}(v) \\
-\frac{1}{3}L\,(2\,\text{Cdy}\,h\,(L^2r^2+3\,v^2)+b\,\text{Cdz}\,((2q^2-r^2)L^2-3\,v^2+6\,w^2))\,\text{sgn}(w)
\end{pmatrix}$$

Appendix 2: Drag Integral Computations

Below is the explicit integral for sway drag of Eq. (3) and (4):

```
Ucf = √((v+ x* r)^2 + (w - x* q)^2);
DragKernel = (Cdyh (v+ x* r)^2 + Cdz b (w - x* q)^2) / Ucf^2;
SwayDrag = Simplify[
  Integrate[DragKernel * (v + x * r) * Ucf, {x, Xnose, Xtail}]];
SwayDrag1 =
 Simplify[SwayDrag /. {Xtail -> -L, Xnose -> L}] // TraditionalForm
```

$$
\begin{aligned}
&\frac{1}{6}\Bigg(\frac{1}{(q^2+r^2)^{7/2}}\Bigg(3q(b\,\mathrm{Cdz}(q^2-4r^2)+\mathrm{Cdy}\,h(3r^2-2q^2))\\
&\quad\log\Bigg(2\Bigg(\frac{L(q^2+r^2)+rv-qw}{\sqrt{q^2+r^2}}+\sqrt{(Lr+v)^2+(w-Lq)^2}\Bigg)\Bigg)(qv+rw)^3\Bigg)-\\
&\frac{1}{(q^2+r^2)^{7/2}}\Bigg(3q(b\,\mathrm{Cdz}(q^2-4r^2)+\mathrm{Cdy}\,h(3r^2-2q^2))\\
&\quad\log\Bigg(2\Bigg(\frac{-L(q^2+r^2)+rv-qw}{\sqrt{q^2+r^2}}+\sqrt{(v-Lr)^2+(Lq+w)^2}\Bigg)\Bigg)(qv+rw)^3\Bigg)+\\
&\frac{1}{(q^2+r^2)^3}\Big(\sqrt{(Lr+v)^2+(w-Lq)^2}\,\big(b\,\mathrm{Cdz}(-L(2Lr+3v)q^6+(Lr+3v)wq^5+\\
&\qquad r(-4L^2r^2-Lvr+13v^2+w^2)q^4+r^2(7Lr+27v)wq^3-\\
&\qquad 2r^3(L^2r^2-Lvr+v^2-5w^2)q^2+6r^4(Lr-v)wq-6r^5w^2)-\\
&\quad \mathrm{Cdy}\,h\,r(18v^2q^4+27rvwq^3+\\
&\qquad 5r^2v^2q^2+11r^2w^2q^2-3r^3vwq+2L^2r^2(q^2+r^2)^2+\\
&\qquad 2r^4v^2-4r^4w^2+Lr(q^2+r^2)(9vq^2+5rwq+4r^2v))\big)\Big)-\\
&\frac{1}{(q^2+r^2)^3}\Big(\sqrt{(v-Lr)^2+(Lq+w)^2}\,\big(b\,\mathrm{Cdz}(L(3v-2Lr)q^6+(3v-Lr)wq^5+\\
&\qquad r(-4L^2r^2+Lvr+13v^2+w^2)q^4+r^2(27v-7Lr)wq^3-\\
&\qquad 2r^3(L^2r^2+Lvr+v^2-5w^2)q^2-6r^4(Lr+v)wq-6r^5w^2)-\\
&\quad \mathrm{Cdy}\,h\,r(18v^2q^4+27rvwq^3+5r^2v^2q^2+\\
&\qquad 11r^2w^2q^2-3r^3vwq+2L^2r^2(q^2+r^2)^2+2r^4v^2-\\
&\qquad 4r^4w^2-Lr(q^2+r^2)(9vq^2+5rwq+4r^2v))\big)\Big)\Bigg)
\end{aligned}
$$

Notice the singularity at the origin. Here is the computation for the approximate formula:

```
In[3]:= SwayDrag = Simplify[
      Integrate[DragKernel * (v + x * r) * Ucf, {x, Xnose, Xtail}]];
    SwayDrag0 = Simplify[SeriesExpansion[SwayDrag, {q, v, w}, 3]];
    SwayDrag1 = (Numerator[SwayDrag0] /.
        {Sqrt[r^2* Xnose^2] -> Sqrt[Abs[r] ^2* Xnose^2],
         Sqrt[r^2* Xtail^2] -> Sqrt[Abs[r] ^2* Xtail^2]}) /
      (Denominator[SwayDrag0] /.
        {Sqrt[r^2* Xnose^2] -> Sqrt[Abs[r] ^2* Xnose^2],
         Sqrt[r^2* Xtail^2] -> Sqrt[Abs[r] ^2* Xtail^2]});
    SwayDrag2 = Cancel[PowerExpand[Simplify[Expand[SwayDrag1]]]];
    SwayDrag3 =
     Simplify[SwayDrag2 /. {Xtail -> -L, Xnose -> L}] // TraditionalForm
```

Out[3]//TraditionalForm=

$$
\frac{1}{3}L(\mathrm{Cdy}\,h((q^2-2r^2)L^2-6v^2+3w^2)-2b\,\mathrm{Cdz}(L^2q^2+3w^2))\,\mathrm{sgn}(r)
$$