Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

On The Modeling of Robots Operating on Ships

Lonnie J. Love, John F. Jansen, and Francois G. Pin
Robotics and Energetic Systems
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6305
lovelj@ornl.gov

Abstract— **The decrease in manpower and increase in material handling needs on many Naval vessels provides the motivation to explore the modeling and control of Naval robotic and robotic assistive devices. This paper presents a simple methodology to symbolically compute the dynamic equations of motion of a serial link manipulation system operating on the moving deck of a ship. First we provide background information that quantifies the motion of the ship, both in terms of frequency and amplitude. We then formulate the motion of the ship in terms of homogeneous transforms. Likewise, the kinematics of a manipulator is considered as a serial extension of the ship motion. We then show how to use these transforms to formulate the kinetic and potential energy of the arm moving on a ship. As a demonstration, we consider two examples: a one degree-of-freedom system experiencing three sea states operating in a plane to verify the methodology and a 3 degree of freedom system experiencing all six degrees of ship motion to illustrate the ease of computation and complexity of the solution. We provide a preliminary comparison between conventional linear control and repetitive learning control (RLC) and show how fixed time delay RLC breaks down due to the varying wave disturbance frequency.**

Keywords-ship motion, modelling, dynamics

I. INTRODUCTION

While there has been ample research directed towards the design and control of surface vessels, underwater manipulators and vehicles, there has been surprisingly little effort devoted toward the design and control of robotic manipulation systems operating on a ship experiencing heavy sea states. A recent exception is crane control on ships [1,2,3,4]. The general problem addressed with this research is that wave induced motion of a ship produces a low frequency disturbance on the motion of a crane. A robotic system, operating under motion or force control on a ship, will likewise experience low frequency disturbances that can impact the precision and performance of the machine. There is a growing need for robotics in the Navy. Reduction in personnel with no tolerance for reduction in capabilities requires increased levels of both machine automation as well as advanced human assistive devices. One example was the development of the Next Generation Munitions Handler (NGMH) by ORNL for the armed services [5]. Traditionally, bomb-loading crews consist of anywhere from four to eight personnel to load bombs and missiles ranging from 500 to 2000 pounds. The current technology is based on crude machines with names such as the "jammer" and "hernia bar". With existing and projected future reduction in military workforce, the armed services are exploring the development of technologies that enable fewer personnel to accomplish the same tasks in the less time. There are a wide variety of potential applications of robotic and human assistive systems on a ship (munitions handling, maintenance, damage control, material handling to name a few). However, the nature of the environment provides a host of unique problems. Specifically, the environment in which the robot operates is continually moving. The motion of the ship generates low frequency disturbance forces on the system, both in terms of inertial forces as well as shifts in the direction of gravity. Subsequently, there is a need for the development of advanced control methodologies to compensate for sea state disturbances. However, the nature of the environment makes it difficult to experimentally test competing control methodologies in a laboratory setting. There are only a handful of sea state simulation platforms that have the capacity to hold a moderately sized robotic system. However, much can be gained by having a high fidelity numerical simulation of a robot that includes ship motion disturbances.

The objective of this paper is to formulate a simple and efficient methodology to derive the dynamic equations of motion for a multi-degree of freedom manipulator moving on a six degree of freedom platform. We begin in Section II with a basic description of models used to describe a variety of sea states. This is followed in Section III by an abbreviated analysis of the motion of a marine vessel experiencing wave loading. The results of this section provide some relevant information quantifying the amplitude and frequency of disturbances expected for a variety of vessels operating under various sea states. In Section IV, we show how the classic homogeneous transform, combined with the energy approach can easily be configured to symbolically calculate the dynamic equations of motion. We model the ship motion as a six-degree of freedom system. The manipulation system is coupled serially to the ship model. We then show how elements of the homogeneous transform can be used to symbolically compute the position, velocity, angular position, and angular velocity vectors of the center of mass of each link and the payload of the manipulator. This basic methodology is applicable to any symbolic computation program. However, we use Matlab® and the Symbolic Toolbox and show through two examples how the resulting symbolic equations of motion can be easily integrated with Simulink® to provide a numerical simulation of the system. The motivation for this work is to develop a platform for testing control algorithms and alternative designs for ship-board robotic and human assistive machines.

0-7803-8232-3/04/$17.00 ©2004 IEEE

II. WAVE MOTION

We begin by considering the source of our problem, wave motion. There has been considerable effort devoted towards the modeling and analysis of wave motion. An irregular wave pattern can be generated through a combination of sinusoidal waves of different amplitudes and frequencies. Since standard waves are characterized as a combination of wave amplitudes and frequencies, it is standard practice to model wave motion as an energy spectrum. The actual units of the spectral model are normalized with respect to water density and gravity, thus the units are in terms of displacement squared over frequency. One popular model is the two-parameter Bretschneider wave spectral model used to define the frequency content of random sea waves. The two parameters are, by definition, the significant wave height ($H_{1/3}$) in centimeters, and the modal wave period (T) in seconds. This significant wave height is defined as the average height of the top 1/3 highest waves. The wave spectral density, $S(\omega_w)$, is defined in Equation (1).

$$S(\omega) = \frac{A}{\omega^5} e^{-B/\omega^4} \text{ with } A = 173 H_{1/3}^2 / T^4 \text{ and } B = 691/T^4 \quad (1)$$

It is common to describe the wave conditions, which include $H_{1/3}$ and T, in terms of a specific "sea state". Table 1 provides a condensed version of the relevant data from the popular Pierson-Moskowitz sea condition definition [6]. Figure 1 shows a representative spectrum for sea state 4.

Table 1: Sea State Definition

Sea State	Description	Wind velocity (knots)	Significant wave height (ft)	Average period (sec)
0	Ripples with appearance of scales, without foam crests	0-2	0-0.01	0.5
1	Small wavelets, short but pronounced crests do not break	5-8.5	0.5-1.3	1.3-2.3
2	Small waves becoming larger	10-13.5	1.8-3.3	2.7-3.6
3	Fairly frequent "white horses"	14-16	3.6-4.7	3.8-4.3
4	Moderates waves, taking a more pronounced long form.	18-20	5-6.6	4.8-5.1
5	Large waves begin to form.	22-24	7.3-10.5	5.4-6.4
6	White crests are more extensive everywhere.	25-28	10.9-14.3	6.6-7.5
7	Sea heaps up and white foam from braking waves blown in streaks along wind direction	30-40	16.4-29.1	8.0-10.7
8	High waves. Sea begins to roll. Visibility affectes.	42-54	40.8-67.4	11.3-14.5
9	Rolling of the sea becomes heavy and shocklike.	>54	>72.5	16-17.2

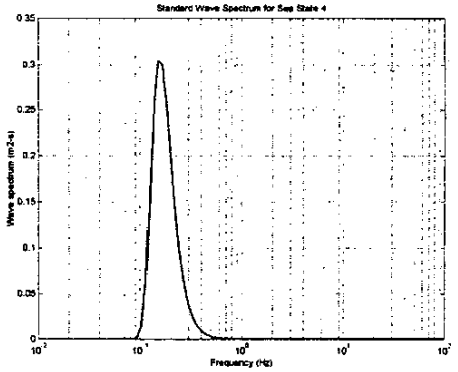

Figure 1: Sea State Spectrum

The distribution of wave spectral energy as a function of ship heading is considered either to be unidirectional (long crested) or spread ±90° about a predominant direction (short crested). Equation (2) accounts for the spread of the energy, transforming from long crested to short crested wave models,

$$S(\omega,\nu) = (2/\pi)\cos^2(\nu - \mu)S(\omega) \quad (2)$$

where μ is the predominant heading of the wave front containing the principal amount of energy and a ν represents the variation of wave energy as a function of the predominant direction of the wave front. There are a variety of methods available for the computation of the wave amplitude time history. It is common practice to quantize the above energy spectrum into N equally spaced elements. The amplitude, ζ_k, at the discrete frequency ω_k, is extracted from the spectral energy in Equation (1) and Figure 1. Equation (3) is the discretized expression for long crested waves

$$\zeta_{LC}(t) = \sum_{k=1}^{N} \zeta_k \cos(\omega_{Ek} t + \gamma_k) \quad (3)$$

where ω_{Ek} is the encountered wave frequency and γ_k is a random phase angle. The encountered wave frequency is actually a doppler shift in the wave frequency, ω, as a function of ship speed (V) and heading (μ).

$$\omega_{Ek} = \left|\omega - (\omega^2 V/g)\cos(\mu)\right| \quad (4)$$

The component amplitude, ζ_k, is computed by

$$\zeta_k = \sqrt{2\int_{\omega_k - d\omega/2}^{\omega_k + d\omega/2} S_\zeta(\omega) d\omega} \quad (5)$$

The number of frequencies, N, used to compute the wave time history should be large enough to obtain a representative Raleigh distribution of single amplitudes. Likewise, the increment in frequency, dω, is equal to the range of frequencies (ω_{max}-ω_{min}) divided by N where ω_{max} and ω_{min} are based upon the frequency range that provides ample representation of the total wave energy. The computation of short crested wave time history is slightly more complex, accounting for angular spread in wave energy. The above description provides the background necessary for the computation of wave time histories.

III. Ship Motion Due To Sea State

There has been a great deal of effort devoted towards modeling ship motion due to wave loadings. However, the primary focus had been directed towards ship design [7,8,9]. Our motivation for understanding ship motion is to quantify the expected magnitude and frequency of disturbance loads for a motion and/or force controlled manipulation system. Subsequently, this section will provide an abbreviated explanation of one of the techniques presently used to model ship motion.

Figure 2 shows a simplified model of a ship with the corresponding displacements due to wave motion. The motion of the ship is defined by six displacements (surge, sway, heave, roll, pitch, and yaw) at the ship's longitudinal center of gravity, from which motions at all other locations on the ship can be developed. While there are a number of techniques to simulate ship motion, the strip theory of Salvensen et al. is one of the more popular approaches to modeling the 6 DOF response for a ship advancing at a constant forward speed with arbitrary heading in regular sinusoidal waves [10]. In it's simplest form, a ship acts as a set of filters, called the Response Amplitude Operators (RAOs), that transforms wave motion into the six degrees of motion (surge, sway, heave, roll, pitch and yaw). Each motion has its own characteristic RAO. As illustrated in the previous section, there is ample information for characterizing the frequency content of the waves. The challenge is to design accurate models of the ship that faithfully characterizes the behavior of the ship. Strip theory is able to provide reliable estimates of sea keeping performance for a wide range of hull forms and sea conditions. Calculations are made in the frequency domain with the warping of the excitation frequency accounting for forward speed and heading, Equation (4). There are three main stages to computing the motion response of the ship. First, divide the ship into a number of transverse sections (or strips), generally from 10 to 40, and compute the two-dimensional hydrodynamic coefficients such as added mass, damping, wave excitation and restoring force. Next, integrate these values along the length of the vessel to obtain global coefficients for the coupled motion of the vessel. Finally, the equations of motion for the ship can be solved to give the amplitudes and phases of the heave, surge, sway, yaw, pitch and roll motions. Clearly, the motion of a ship is a complex phenomenon and the above description is merely a simplified explanation of one method used for modeling ship motion. The above description is intended to only provide insight into the problem of ship motion simulation. The interested reader is referred to the following list of articles and text for a deeper understanding of ship motion simulation [11,12,13]. Fortunately, there are a number of commercial software packages available for the analysis and simulation of marine vessels. The level of sophistication, as well as magnitude of cost, varies dramatically. The package used for the analysis in the paper is the Simulation Time History (STH) and Access Time History (ACTH) programs developed at the Naval Surface Warfare Center in Bethesda Maryland and are available through the National Technical Information Service (NTIS). Table 2 provides a general description of the expected motion of a ship based upon the sea state and vessels length and beam dimensions [15]. A listing of naval and commercial vessels with their respective displacement, length and beam dimensions follows in Table 3. A full listing of the data on each of the above navy war ships is available through a navy web site [16].

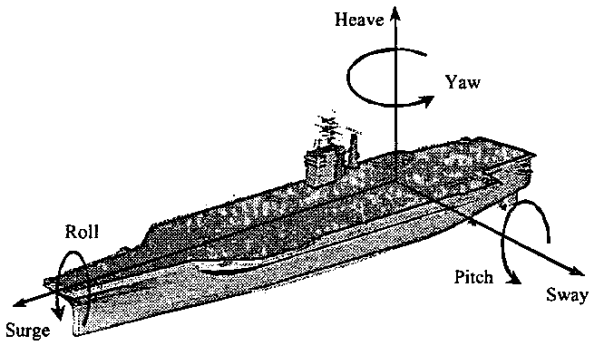

Figure 2: Ship Displacements

Table 2: Ship Response as function of Sea State

Sea State	Wave Ht (ft)	Boat Length (ft)	Pitch (deg)	Pitch Period (sec)	Heave (ft)	Heave Acc (g)	Beam (ft)	Roll Angle (deg)	Roll Period (sec)
4	6.8	<150	2	3.5	2.2	0.10	<50	7	7.1
		150-250	2	4	2.2	0.10	50-75	6	11.5
		250-350	1	5	2.2	0.10	75-105	6	13.7
		350-500	1	6	1.7	0.08	>105	5	14.8
		500-700	1	7	1.3	0.06			
		>700	1	8	0.9	0.04			
5	9.8	<150	3	3.5	5.2	0.17	<50	12	7.1
		150-250	3	4	5.2	0.17	50-75	10	11.5
		250-350	2	5	5.2	0.17	75-105	10	13.7
		350-500	2	6	4.3	0.14	>105	9	14.8
		500-700	2	7	3.1	0.10			
		>700	1	8	2.1	0.07			
6	17	<150	5	3.5	15.0	0.27	<50	19	7.1
		150-250	4	4	15.0	0.27	50-75	16	11.5
		250-350	4	5	15.0	0.27	75-105	15	13.7
		350-500	3	6	11.7	0.21	>105	13	14.8
		500-700	3	7	8.9	0.16			
		>700	2	8	6.1	0.11			

Table 3: Ship Size/Displacement

Type of Ship	Displacement (tons)	Length (ft)	Beam (ft)
Aircraft Carrier	97,000	1092	134
Queen Elizabeth II	66,000	887	103
Battleship	59,000	860	108
Amphibious Assault	40,500	844	106
Ammunition ship	18,000	564	81
Command Ship	14,640	520	84
Cruiser, Destroyer	9600	567	55
Frigates	4100	445	45
Rescue, Salvage	3280	255	51
Coastal Patrol	331	170	33

IV. Dynamic Equations of Motion

At this point, we have an ability to model the motion of a ship as a function of the sea state and vessel. We now are interested in understanding the impact this ship motion has on the dynamics of a general manipulation system. Our approach to modeling the dynamics of a robot on a moving platform, such as a ship, consists of: modeling the ship motion and robot kinematics with homogeneous transforms, constructing kinetic and potential energy terms using these transforms, and

symbolically computing the dynamic equations of motion via the Lagrange approach. First, as a review, the homogeneous transform is expressed using the traditional Denavit-Hartenberg (D-H) representation found in most robotics texts where the four quantities θ_i (angle), α_i (twist), d_i (offset), a_i (length) are parameters of link and joint i[17, 18].

$$H_i = \begin{bmatrix} c_{\theta i} & -s_{\theta i}c_{\alpha i} & s_{\theta i}s_{\alpha i} & a_i c_{\theta i} \\ s_{\theta i} & c_{\theta i}c_{\alpha i} & -c_{\theta i}s_{\alpha i} & a_i s_{\theta i} \\ 0 & s_{\alpha i} & c_{\alpha i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

The conventional use of the homogeneous transform treats each subsequent transformation as a body fixed rotation and translation. However, the sea state is generally described in terms of a space fixed displacements. e.g. all of the translations and rotations are with respect to the same space fixed reference frame. Referring to Figure 2, roll and surge are about a fixed X-axis, pitch and sway are about a fixed Y-axis and yaw and heave are about a fixed Z-axis. We begin by constructing a homogeneous transformation using space-fixed rotations and translations for a transformation from the sea coordinate frame to the base of the robot. Equation (7) is the final expression for the displacement of the base of a robot with respect to a sea state where $c\theta$ is $\cos(\theta)$ and $s\theta$ is $\sin(\theta)$.

$$H_{sea}^{base} = \begin{bmatrix} c_{\theta pitch}c_{\theta yaw} & c_{\theta pitch}s_{\theta yaw} & -s_{\theta pitch} & Surge \\ s_{\theta roll}s_{\theta pitch}c_{\theta yaw} - c_{\theta roll}s_{\theta yaw} & s_{\theta roll}s_{\theta pitch}s_{\theta yaw} + c_{\theta roll}c_{\theta yaw} & s_{\theta roll}c_{\theta pitch} & Sway \\ c_{\theta roll}s_{\theta pitch}c_{\theta yaw} + s_{\theta roll}s_{\theta yaw} & c_{\theta roll}s_{\theta pitch}s_{\theta yaw} - s_{\theta roll}c_{\theta yaw} & c_{\theta roll}c_{\theta pitch} & Heave \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

We assume for now that the sea states are defined with respect to the base of our robot. If necessary, additional transformations can be included from the coordinate system of the sea state to the base of the manipulator. We also assume that we can define homogeneous transforms from each joint to a point on each link where the associated mass properties (mass and inertia matrix) are known. So, our basic methodology consists of using the homogenous transforms to identify the displacements and velocities, both translation and rotation, of the center of mass of each link and payload with respect to the manipulators state and the sea state. We extract out of the transforms the vertical displacement of each center of mass for an expression of the total potential energy of the system. Likewise, computation of the system's kinetic energy is based on computing the linear and angular velocity of each link center of gravity with respect to the inertial frame. Once the kinetic and potential energy terms are derived, we simply use the jacobian() function to symbolically calculate the mass matrix and nonlinear dynamic terms following the Lagrange formulation.

First, the position of the center of mass for each link, with respect to the system's inertial coordinate system, is computed by post multiplying the homogeneous transform from the robot base to the link's c.g. by H_{sea}.

$$\begin{aligned} H^i &= H_{sea}^{base} H_{base}^i \\ &= \begin{bmatrix} R_{sea}^{base} & \bar{x}_{sea}^{base} \\ \bar{0} & 1 \end{bmatrix} \begin{bmatrix} R_{base}^i & \bar{x}_{base}^i \\ \bar{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_{sea}^i & \bar{x}_{sea}^i \\ \bar{0} & 1 \end{bmatrix} \end{aligned} \tag{8}$$

The potential energy due to gravity for link i is the vertical component (z_{sea}^i in direction of gravity) of $\bar{x}_{sea}^i$ times the mass of the link.

$$V^i = m_i g z_{sea}^i \tag{9}$$

To compute the kinetic energy, we must first derive expressions for the linear and angular velocity of the center of gravity for each link as a function of the sea state and states of the manipulator. We have an expression, $\bar{x}_{sea}^i$ in Equation (8), for the position of the c.g. of link i with respect to the sea inertial frame. The velocity vector, $\bar{v}_i$, is computed by multiplying the Jacobian (with respect to the combined states of the manipulator $\bar{q}$) of $\bar{x}_{sea}^i$, $J(\bar{x}_{sea}^i, \bar{q})$, by the state velocity vector, $\dot{\bar{q}}$.

$$\begin{aligned} \bar{v}_i &= \frac{\partial \bar{x}_{sea}^i}{\partial t} \\ &= \sum_{j=1}^{6+i} \frac{\partial \bar{x}_{sea}^i}{\partial q_j} \dot{q}_j \\ &= J(\bar{x}_{sea}^i, \bar{q})\dot{\bar{q}} \end{aligned} \tag{10}$$

where $\bar{q} = [\bar{q}_{sea}\ \bar{q}^i]$

with $\bar{q}_{sea} = [roll, pitch, yaw, surge, sway, heave]$

and $\bar{q}^i = [q_0 q_1 \ldots q_i]$

The rotational velocity is a little more involved, but can be simplified by again using the homogeneous transform and starting at the base of the robot and working forward to the c.g. of each link. We begin by defining the base rotational velocity.

$$\bar{\omega}_{base} = \begin{bmatrix} \dot{\theta}_{roll} & \dot{\theta}_{pitch} & \dot{\theta}_{yaw} \end{bmatrix} \tag{11}$$

We combine the rotational velocity of the first link (with respect to the link), $\dot{\bar{q}}_1$, with the projection of ω_{base} to the center of mass of the link, again using the rotational component of the homogeneous transform in Equation (7).

$$\bar{\omega}_1 = \dot{\bar{q}}_1 + R_{base}^1 \bar{\omega}_{base} \tag{12}$$

Each subsequent joint consists of projecting the total angular velocity vector of the previous joint to the current joint's coordinate system, using the rotational component of that joint's homogenous transform, and adding the joint angular velocity.

$$\bar{\omega}_i = \dot{\bar{q}}_i + R_{i-1}^i \bar{\omega}_{i-1} \tag{13}$$

We now have expressions for the linear and angular velocity of the center of mass for each link. The total kinetic energy of the system is

$$T = \frac{1}{2}\sum_{i=1}^{N} m_i v_i^t v_i + R_{base}^i \bar{\omega}_i^t [I_i] \bar{\omega}_i \tag{14}$$

where m_i is the mass of link i and I_i is the inertia matrix of link i about the center of gravity. As a final step, we add external forces applied to the system. For now, we assume forces are applied only to the joints and tip of the robot. We use the principle of virtual work to lump these terms together.

$$\begin{aligned} \partial W &= \sum_{i=1}^{N} \tau_i \partial q_i + \bar{F}_{tip} \partial \bar{x}_{tip} + \bar{M}_{tip} \partial \theta_{tip} \\ &= \left[\bar{\tau} + J_{tip}^t(\bar{q}) \begin{Bmatrix} \bar{F}_{tip} \\ \bar{M}_{tip} \end{Bmatrix} \right] \partial \bar{q} \\ &= \bar{Q} \partial \bar{q} \\ J_{tip}(\bar{q}) &= \frac{\partial \bar{x}_{base}^{tip}}{\partial \bar{q}} \end{aligned} \tag{15}$$

Equations (9) and (14) provide expressions for the kinetic and potential energy of the system. We start with the classic definition of the Lagrange equations of motion.

$$\frac{\partial}{\partial t}\left(\frac{\partial T}{\partial \dot{\bar{q}}}\right) - \frac{\partial T}{\partial q} + \frac{\partial V}{\partial q} = \bar{Q} \tag{16}$$

The first term in Equation (16) can be expanded using the chain rule.

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \bar{q}}\frac{\partial \bar{q}}{\partial t} + \frac{\partial}{\partial \dot{\bar{q}}}\frac{\partial \dot{\bar{q}}}{\partial t} \tag{17}$$

Substituting Equation (17) into (16),

$$\frac{\partial}{\partial \dot{\bar{q}}}\left(\frac{\partial T}{\partial \dot{\bar{q}}}\right)\ddot{\bar{q}} + \frac{\partial}{\partial \bar{q}}\left(\frac{\partial T}{\partial \dot{\bar{q}}}\right)\dot{\bar{q}} - \frac{\partial T}{\partial \bar{q}} + \frac{\partial V}{\partial \bar{q}} = \bar{Q} \tag{18}$$

As with the velocity computation in Equation (10), we can exploit the jacobian() function in Matlab® for the evaluation of many of the terms in Equation (18). First, the term $\partial T/\partial \dot{\bar{q}}$ is the differential of the scalar kinetic energy term with respect to the full state velocity vector defined in Equation (10). This results in the vector, Lv, in the script files in a technical report.[18] We then take the Jacobian on Lv with respect to the full state vector and full state velocity vector to the first and second terms in Equation (18). Likewise, the 3rd and 4th terms in Equation (18) are evaluated using the Jacobian function on the kinetic and potential energy terms respectively. Thus, it should be clear that once the kinetic and potential energy terms are defined, it is straightforward to symbolically evaluate the dynamic equations of motion using Matlab's jacobian() function. The Jacobian for projecting external forces to the generalized coordinates can similarly be computed using the tip position of the robot and the jacobian function.

We provide two examples: a simple one degree of freedom system operating in a plane and a three degree of freedom system experiencing all six degrees of motion from the sea state. The first example is simple enough to verify through hand calculations. The second example is more complex, yet practical. Figure 3 shows the basic kinematic model of the one degree of freedom system experiencing 3 sea states in the X-Y plane. We are assuming a one DOF system with mass M and rotary inertia I_z located at the tip of a link of length L. The system is experiencing only three of the six sea states: surge (x_s), heave (y_s), and pitch (θ_s). The only external force applied to the system is a joint torque, τ, applied at joint 1. The results are displayed in Equation (19).

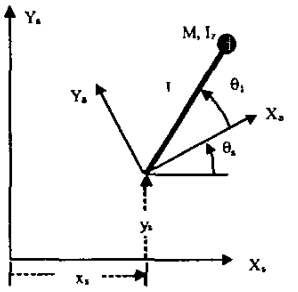

Figure 3: One DOF Model

$$\begin{aligned} &(M * L1^2 + I_{1z})(\ddot{\theta}_1 + \ddot{\theta}_p) + M * L1 * \cos(\theta_p + \theta_1)\ddot{y}_s - \\ &M * L1 * \sin(\theta_p + \theta_1)\ddot{x}_s + M * g * L1 * \cos(\theta_p + \theta_1) = \tau_1 \end{aligned} \tag{19}$$

The power of this approach is more evident as we progress to more complex systems. Additional degrees of freedom only require additional homogeneous transforms. As a second example, we derive the dynamic equations of motion for the 3 degree of freedom system, shown in Figure 4, with the full six degrees of freedom from the sea state. A listing of the code used for computing the dynamics of the strength amplifying machine on the deck of a ship is shown in the listing in Appendix A. The resulting equations of motion can be partitioned into a compact form, Equation (20),

$$\begin{bmatrix} M_{rr} & M_{rs} \\ M_{rs}^t & M_{ss} \end{bmatrix} \begin{Bmatrix} \ddot{\bar{q}}_r \\ \ddot{\bar{q}}_s \end{Bmatrix} + \begin{Bmatrix} NLT_r \\ NLT_s \end{Bmatrix} = \begin{Bmatrix} \bar{Q}_r \\ \bar{0} \end{Bmatrix} + \begin{Bmatrix} J^t(q_r)F_{ext} \\ \bar{0} \end{Bmatrix} \tag{20}$$

where M_{rr} is the 3x3 mass matrix for the robot with respect to the robot's state acceleration, M_{ss} is the 6x6 mass matrix of the robot with respect to the sea state acceleration, NLT_r is a 3x1 vector of the nonlinear terms (gravitational, coriolis, centripetal) as a function of both the robot's state and the sea state and Q_r is the joint force input to the system, F_{ext} is an external force vector applied to the end effector and $J^t(q_r)$ is the Jacobian from the end effector to the joint space. In order to include the dynamic equations of motion in Simulink®, we use Equation (21) to solve for the acceleration of the robot's state vector as a function of all of the inputs (external forces and joint torques), system state (position and velocity) and external disturbances (sea state position, velocity, and acceleration).

$$\ddot{\bar{q}}_r = M_{rr}^{-1}\left\{\bar{Q}_r + J^t(q_r)F_{ext} - NLT_r - M_{rs}\ddot{\bar{q}}_s\right\} \tag{21}$$

Figure 4: Strength Amplifying Machine

Appendix A provides a listing of the Matlab® code required to symbolically compute the dynamic equations of motion for the system displayed in Figure 4. While the output of the single degree of freedom, planar case can be listed in Equation (19), the results of the dynamic equations of motion for the second system generates 84 pages of c-code and would require considerable effort to derive by hand. The first obvious question is the validity of the results. For now, we can only verify the basic methodology by comparing to simple cases. To date, we have compared the methodology to a number of manipulators with stationary bases and achieve the same symbolic results. In addition, we have considered only simple one and two degree of freedom systems experiencing one to three degrees of ship motion. In each case, the symbolic solutions are the same leading us to believe the methodology is sound. The second obvious question is what can you do with 84 pages of c-code. Fortunately, the code can be directly imported into Simulink® through the S-Function builder. Finally, one might ask how long does it take to simulate a system with 84 pages of c-code. We will conclude the paper with some simulation results of our 3 degree of freedoms system experiencing 6 dof of ship motion. The simulation includes nonlinear dynamic modeling of the hydraulic system (servo-valve orifice equations, asymetric cylinders, fluid compliance…), controls and the dynamic equations of motion computed above. The simulation was surprisingly fast. It takes 178 seconds to execute a 120 second simulation with a fixed 0.01 second time step and 4th order Runge-Kutta integration, executed on a 750 MHz Pentium III laptop. The motivation for computing the dynamics equations of motion are two fold. First, by having the dynamics in a symbolic form, it is possible to aid in the design process, changing parameters to optimize the system. Second, a model of the system dynamics can aid in increasing the fidelity of simulation for control design and analysis. As a concluding exercise, we demonstrate the dynamic model through simulation. First, we have formulated the dynamic equations of motion so as to compute the joint accelerations as a function of the sea state, external forces applied to the tip of the arm, and joint torques, Equation (21). The arm in Figure 5 is hydraulically actuated. The hydraulic actuator models generate force as a function of the servovalve current, actuator position and velocity.

For our first simulation, we start with a sinusoidal models with a fixed frequency for each of the six sea states. We assume a significant wave height of 7 ft and average period of 6 seconds (sea state 5). For demonstation purposes, the system has a 500 lb payload, has linear position control with a gain margin of 10 dB and phase margin of 60 degrees, and is commanded to stay at a specific position. Under these sea states, the vertical and horizontal tracking error exceed 1 inch. We then introduce a Repetitive Learning Controller (RLC), Figure 5, with a fixed delay (T_d) that is the same as our simulated wave period. Details on the design of an RLC (specifically the filters q(s) and b(s)) can be found elsewhere in the literature.[19] The results in Figures 6 and 7 show a reduction of the vertical and horizontal position error by over an order of magnitude with the introduction of the RLC.

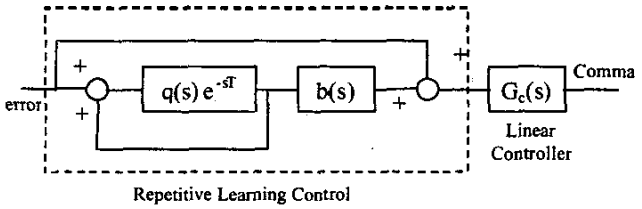

Figure 5: Repetitive Learning Control

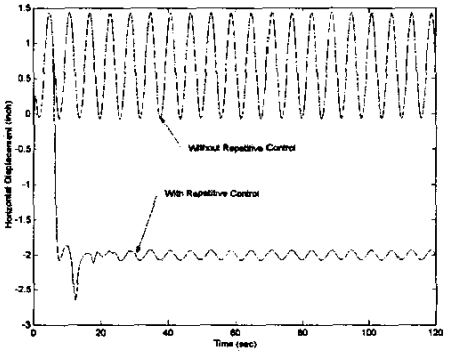

Figure 6: Horizontal Response with 500 lb Payload

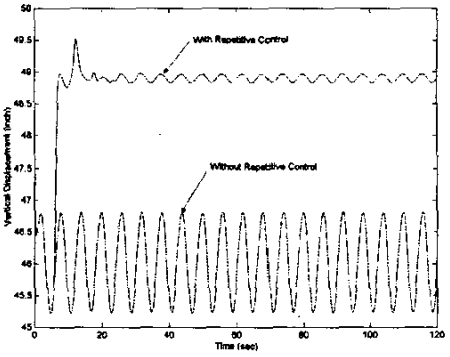

Figure 7: Vertical Response with 500 lb Payload

We now repeat the same series of simulations but introduce more realistic wave and sea state disturbances, generated for the same sea state as in Figures 6 and 7, but using the STH and ACTH programs. Figures 8 and 9 compare the horizontal and vertical position of the arm, with the RLC disabled and

enabled. The delay time for the RLC was selected based on the significant wave period. Clearly, the variations in the wave period negatively impact the performance of the fixed time delay RLC. There are some time segments (from 90 to 120 seconds) where it appears there is some benefit to using RLC. These results provide the motivation for an exploration of ship motion compensation for robotic systems. We anticipate that the impact will be more dramatic for force controlled devices. The interested reader can find greater details in a report covering the modeling, simulation and control of maritime manipulation systems.[18]

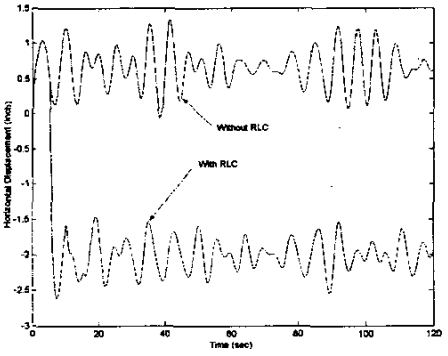

Figure 8: Horiz. Response, Realistic Waves, 500 lb Payload

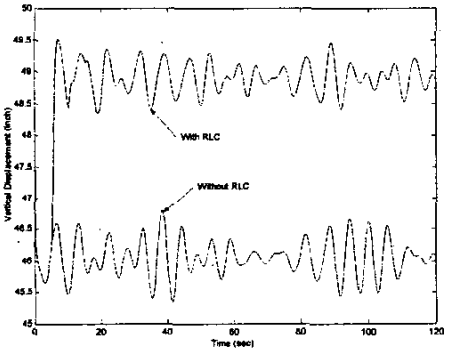

Figure 9: Vertical Response, Realistic Waves, 500 lb Payload

V. Conclusions and Results

This article has described the fundamental problems associated with motion control of manipulation systems operating on the deck of a moving ship. We provided a brief survey of present wave modeling techniques and ship motion simulation procedures. This is followed by a methodology to compute the dynamic equations of motion, using energy methods, of a general serial link manipulator on a six degree of freedom base. We provide as an example a three degree of freedom manipulator and show, through simulation, the impact of wave generated disturbance on the tracking control of this system. Subsequent work is exploring adaptive learning control methodologies for compensating for the time varying periodic nature of these disturbances. In addition, we are exploring the impact wave dynamics have on force controlled manipulation systems. Here, the wave disturbance has a more dramatic impact with direct feedback of the force in the control algorithm.

Acknowledgements

The authors would like to acknowledge the support of Dr. Theresa McMullen and the Office of Naval Research under Interagency Agreement No. 1866-Q356-A1, whose support made this work possible.

VI. Appendix A

```
syms ai alfai di thi mass g
syms q1 q1_d q1_dd
syms roll pitch yaw heave surge sway
syms roll_d pitch_d yaw_d heave_d surge_d sway_d
syms roll_dd pitch_dd yaw_dd heave_dd surge_dd sway_dd
syms th1 th1d th1dd th2 th2d th2dd th3 th3d th3dd
syms L1 L2 L3 L4 L1c L2c L3c L3x L3h L3y L4c
syms m1 m2 m3 I1x I2x I3x I1y I2y I3y I1z I2z I3z
pi = sym('pi');

% Symbolically derive motion of base of robot on deck of ship experiencing
% 6 dof of sea motion (roll, pitch, yaw, heave, surge, sway). Use DH parameters
% to describe this motion in terms of homogeneous transforms.
R1s=[1  0         0;
   0  cos(roll)  sin(roll);
   0  -sin(roll) cos(roll)];

R2s=[cos(pitch) 0  -sin(pitch);
   0       1  0;
   sin(pitch) 0  cos(pitch)];

R3s=[cos(yaw)   sin(yaw)  0;
   -sin(yaw)  cos(yaw)  0;
   0        0.       1];

Hsea=[[simple(R1s*R2s*R3s) [surge;sway;heave;]];[0 0 0 1]];

ai=-L1;alfai=pi/2;di=0;thi=th1;
H1=[cos(thi)  -sin(thi)*cos(alfai)   sin(thi)*sin(alfai)   ai*cos(thi);
   sin(thi)  cos(thi)*cos(alfai)   -cos(thi)*sin(alfai)   ai*sin(thi);
   0       sin(alfai)          cos(alfai)          di;
   0       0                   0                   1];

ai=L2;alfai=0;di=0;thi=th2+pi/2;
H2=[cos(thi)  -sin(thi)*cos(alfai)   sin(thi)*sin(alfai)   ai*cos(thi);
   sin(thi)  cos(thi)*cos(alfai)   -cos(thi)*sin(alfai)   ai*sin(thi);
   0       sin(alfai)          cos(alfai)          di;
   0       0                   0                   1];

ai=L2c;alfai=0;di=0;thi=th2+pi/2;
H2c=[cos(thi)  -sin(thi)*cos(alfai)   sin(thi)*sin(alfai)   ai*cos(thi);
   sin(thi)  cos(thi)*cos(alfai)   -cos(thi)*sin(alfai)   ai*sin(thi);
   0       sin(alfai)          cos(alfai)          di;
   0       0                   0                   1];

ai=L3x;alfai=0;di=0;thi=th3-pi/2;
H3=[cos(thi)  -sin(thi)*cos(alfai)   sin(thi)*sin(alfai)   ai*cos(thi);
   sin(thi)  cos(thi)*cos(alfai)   -cos(thi)*sin(alfai)   ai*sin(thi);
   0       sin(alfai)          cos(alfai)          di;
   0       0                   0                   1];

ai=L3h;alfai=0;di=0;thi=th3-pi/2;
H3h=[cos(thi)  -sin(thi)*cos(alfai)   sin(thi)*sin(alfai)   ai*cos(thi);
   sin(thi)  cos(thi)*cos(alfai)   -cos(thi)*sin(alfai)   ai*sin(thi);
   0       sin(alfai)          cos(alfai)          di;
   0       0                   0                   1];

ai=L3c;alfai=0;di=0;thi=th3-pi/2;
H3c=[cos(thi)  -sin(thi)*cos(alfai)   sin(thi)*sin(alfai)   ai*cos(thi);
   sin(thi)  cos(thi)*cos(alfai)   -cos(thi)*sin(alfai)   ai*sin(thi);
   0       sin(alfai)          cos(alfai)          di;
   0       0                   0                   1];

ai=L3y;alfai=0;di=0;thi=pi/2;
H4=[cos(thi)  -sin(thi)*cos(alfai)   sin(thi)*sin(alfai)   ai*cos(thi);
```

```
   sin(thi)  cos(thi)*cos(alfai)   -cos(thi)*sin(alfai)   ai*sin(thi);
   0         sin(alfai)           cos(alfai)             di;
   0         0                    0                      1];

ai=L4;alfai=-pi/2;di=0;thi=th4-pi/2;
H5=[cos(thi)  -sin(thi)*cos(alfai)   sin(thi)*sin(alfai)   ai*cos(thi);
   sin(thi)  cos(thi)*cos(alfai)   -cos(thi)*sin(alfai)   ai*sin(thi);
   0         sin(alfai)           cos(alfai)             di;
   0         0                    0                      1];

H=simple(H1*H2*H3*H4*H5); % Homogeneous transform for the robot
H_full=simple(Hsea*H);   % Full homogeneous transform of robot include sea state

q=[th1;th2;th3];     % state vector of robot
qd=[th1d;th2d;th3d]; % derivitive of state vector
qs=[roll;pitch;yaw;heave;surge;sway]; % sea state vector
qsd=[roll_d;pitch_d;yaw_d;heave_d;surge_d;sway_d]; % sea state velocity
qsdd=[roll_dd;pitch_dd;yaw_dd;heave_dd;surge_dd;sway_dd]; % sea state acceleration

% velocity computation.  Each velocity is the velocity of the cg of the line wrt base
coordinate system
% velocity of cg of 2nd link
Ht=simple(Hsea*H1*H2c);  % homogeneous transform from base to cg of link 2
R2c=Ht((1:3),4);   % pull out x, y, z (vector from base to link 2 cg)
V2c=Jacobian(R2c,[q(1:2);qs])*[qd(1:2);qsd];    % calculate velocity of cg of link 2 wrt
inertial frame (V = dR/dt = dR/dq * dq/dt)

% velocity of cg of 3rd link
Ht=simple(Hsea*H1*H2*H3c);
R3c=Ht((1:3),4);
V3c=simple(Jacobian(R3c,[q;qs])*[qd;qsd]);

% rotation matricies from base to each associated coordinate system
R1=transpose(H1(1:3,1:3));
R2=transpose(H2(1:3,1:3));
R3=transpose(H3(1:3,1:3));

% angular velocity of each link about cg wrt local coordinate frame
Q1=[0;0;th1d]+qsd(1:3);
Q2=simple(R2*R1*Q1+[0;0;th2d]);
Q3=simple([0;0;th3d]+R3*Q2);

% inertia matrix for each link about center of gravity wrt coordinate frame of line (same
as homogeneous transform, translated to cg)
I1=[I1x 0 0;0 I1y 0;0 0 I1z];
I2=[I2x 0 0;0 I2y 0;0 0 I2z ];
I3=[I3x 0 0;0 I3y 0;0 0 I3z];

% Payload information (position/velocity)
syms M_payload;
Rtip=H_full(1:3,4);
Vtip=Jacobian(Rtip,[th1;th2;th3;roll;pitch;yaw;heave;surge;sway])*[th1d;th2d;th3d;roll_
d;pitch_d;yaw_d;heave_d;surge_d;sway_d];

% total kinetic energy:  T = 1/2 qdot' * J * qdot + 1/2 V' M V
T=(1/2*transpose(Q1)*I1*Q1 + 1/2*transpose(Q2)*I2*Q2 + 1/2*transpose(Q3)*I3*Q3 +
1/2*m2*transpose(V2c)*V2c+1/2*m3*transpose(V3c)*V3c)+
1/2*M_payload*(transpose(Vtip)*Vtip);;

% potential energy due to gravity
V=m2*g*R2c(3)+m3*g*R3c(3)+M_payload*g*Rtip(3);

% calculate dT/dqdot
dT_qdot=Jacobian(T,qd);

% extract out mass matrix
MassMatrix= simple(Jacobian(dT_qdot,qd));

% now finish off with remaining terms
NLT1=simple(Jacobian(dT_qdot,[q])*[qd]);
NLT2=simple(Jacobian(dT_qdot,transpose(qs))*qsd);
NLT3=simple(Jacobian(dT_qdot,transpose(qsd))*qsdd);
NLT4=simple(-1*transpose((Jacobian(T,q))));
NLT5=simple(((Jacobian(V,q))));

% translate to C-code
MassMatrix_cc=ccode(MassMatrix);
NLT1_cc=ccode(NLT1);
NLT2_cc=ccode(NLT2);
NLT3_cc=ccode(NLT3);
NLT4_cc=ccode(NLT4);
NLT5_cc=ccode(NLT5);

% calculation of jacobian from tip frame to joint space
LDRDJacobian=simple(Jacobian(H(1:3,4),[th1;th2;th3]));
LDRDJacobian_cc=ccode(LDRDJacobian);
```

VII. REFERENCES

[1] R. J. Henry, Z. N. Masoud, A. H. Nayfeh, and D. T. Mook, "Cargo Pendulation Reduction on Ship-Mounted Cranes Via Boom-Luff Angle Actuation," submitted for publication, Journal of Vibration and Control.

[2] C-M. Chin, A. H. Nayfeh, and D. T. Mook, ``Dynamics and Control of Ship-Mounted Cranes," AIAA Paper No. 98-1731, *39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Long Beach, CA, April 1998.

[3] W. Lacarbonara, R. R. Soper, A. H. Nayfeh, and D. T. Mook, "A Fully Passive Architecture for Pay-Load Pendulation Control in Cranes," *International Conference on Monitoring and Control of Marine and Harbour Structures*, Genoa, Italy, 1999.

[4] B. Kimiaghalam, A. Homaifar, and M. Bikdash, "Feedback and feedforward control law for a ship crane with the Maryland rigging system", Proceedings of *the 2000 ACC.*

[5] T. Deeter, G. Koury, K. Rabideau, M. Leahy, and T. Turner, "The Next Generation Munitions Handler Advanced Technology Demonstrator Program," Proceedings of the 1997 IEEE International Conference on Robotics and Automation, pp. 341-345, 1997.

[6] R. Bhattacharyya, Dynamics of Marine Vehicles, John Wiley & Sons, 1978.

[7] A. Engle, A. Lin, N. Salvensen, and Y. Shin, "Application of 3-D Nonlinear Wave-Load and Structural Response Simulations in Naval Ship Design," *Naval Engineers Journal,* Vol. 109, No.3, May 1997.

[8] M. Abkowitz, "Applications of the Spectral Techniques to Design and Operation," Proceedings of the Society of Naval Architects and Marine Engineers Sea Keeping Symposium, 1973.

[9] T. Lamb, "Organization Theory and Shipbuilding -- A Brief Overview," *Marine Technology*, April 1992.

[10] N. Salvensen, O. Tuck, and O. Faltinsen, "Ship Motions and Sea Loads," *Transactions of the Society of Naval Architects and Marine Engineers*, Vol.78, pp.250-287, 1970.

[11] W. Price, and R. Bishop, Probabilistic Theory of Ship Dynamics, Halsted Press, 1974.

[12] W. Price, Dynamics of Ships, Scholium International, 1991.

[13] T. Ogilvie, "Fundamental Assumptions in Ship-motion Theory," The Dynamics of Marine Vehicles and Structures in Waves, Edited by R.E.D. Bishop and W.G. Price, Institute of Mechanical Engineers, London, 1975, pp.135-145.

[14] DOD-STD-1399, Interface Standard for Shipboard Systems, Section 301A

[15] http://www.chinfo.navy.mil/navpalib/factfile/ffiletop.html

[16] Spong, M., and Viyasagar, M., Robot Dynamics and Control, John Wiley & Sons, 1989.

[17] Yoshikawa, T., Foundations of Robotics: Analysis and Control, MIT Press, 1990.

[18] L.Love, J. Jansen, F. Pin, "Compensation of Wave-Induced Motion and Force Phenomena for Ship-Based High Performance Robotic and Human Amplifying Systems," ORNL/TM-2003/233, October 2003.

[19] K. Srinivasan and F. Shaw, "Analysis and Design of Repetitive Control Systems Using the Regeneration Spectrum," Journal of Dynamic Systems, Measurement, and Control, Vol. 113, pp.216-222, 1991.