

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Unified Dynamics-based Motion Planning Algorithm for Autonomous Underwater Vehicle-Manipulator Systems (UVMS)

Tarun K. Podder*, Nilanjan Sarkar**

*University of Rochester, Rochester, NY, USA

**Vanderbilt University, Nashville, TN, USA

1. Introduction

Among the underwater robotic systems that are currently available, remotely operated vehicles (ROVs) are the most commonly used underwater robotic systems. A ROV is an underwater vehicle that is controlled from a mother-ship by human operators. Sometimes a ROV is equipped with one or more robotic manipulators to perform underwater tasks. These robotic manipulators are also controlled by human operators from a remote site (e.g., mother-ship) and are known as tele-manipulators. Although the impact of ROVs with tele-manipulators is significant, they suffer from high operating cost because of the need for a mother-ship and experienced crews, operator fatigue and high energy consumption because of the drag generated by the tether by which the ROV is connected to the ship. The performance of such a system is limited by the skills, coordination and endurance of the operators. Not only that, communication delays between the master and the slave site (i.e., the mother-ship and the ROV) can severely degrade the performance.

In order to overcome some of the above-mentioned problems, autonomous underwater vehicles (AUVs) are developed. However, an AUV alone cannot interact with the environment. It requires autonomous robotic manipulator(s) attached to it so that the combined system can perform some useful underwater tasks that require physical contact with the environment. Such a system, where one or more arms are mounted on an AUV, is called an autonomous underwater vehicle-manipulator system (UVMS).

One of the main research problems in underwater robotics is how to design an autonomous controller for a UVMS. Since there is no human operator involved in the control of a UVMS, the task planning has become an important aspect for smooth operation of such a system. Task planning implies the design of strategies for task execution. In other words, a task planning algorithm provides a set of desired (i.e., reference) trajectories for the position and force variables, which are used by the controller to execute a given task. Task planning can be divided into motion planning and force planning. In this research, we focus on the design of motion planning algorithms for a UVMS.

The motion planning of a UVMS is a difficult problem because of several reasons. First, a UVMS is a kinematically redundant system. A kinematically redundant system is one which has more than 6 degrees-of-freedom (DOF) in a 3-D space. Commonly, in a UVMS, the AUV has 6 DOF. Therefore, the introduction of a manipulator, which can have n DOF, makes the

combined system kinematically redundant. Such a system admits infinite number of joint space solutions for a given Cartesian space coordinates, and thus makes the problem of motion planning a difficult one. Second, a UVMS is composed of two dynamic subsystems, one for the vehicle and one for the manipulator, whose bandwidths are vastly different. The dynamic response of the vehicle is much slower than that of the manipulator. Any successful motion planning algorithm must consider this different dynamic bandwidth property of the UVMS. There are several other factors such as the uncertainty in the underwater environment, lack of accurate hydrodynamic models, and the dynamic interactions between the vehicle and the manipulator to name a few, which makes the motion planning for a UVMS a challenging problem.

In robotics, trajectory planning is one of the most challenging problems (Klein & Huang, 1983). Traditionally, trajectory planning problem is formulated as a kinematic problem and therefore the dynamics of the robotic system is neglected (Paul, 1979). Although the kinematic approach to the trajectory planning has yielded some very successful results, they are essentially incomplete as the planner does not consider the system's dynamics while generating the reference trajectory. As a result, the reference trajectory may be *kinematically admissible* but may not be *dynamically feasible*.

Researchers, in the past several years, have developed various trajectory planning methods for robotic systems considering different kinematic and dynamic criteria such as obstacle avoidance, singularity avoidance, time minimization, torque optimization, energy optimization, and other objective functions. A robotic system that has more than 6 dof (degrees-of-freedom) is termed as kinematically redundant system. For a kinematically redundant system, the mapping between task-space trajectory and the joint-space trajectory is not unique. It admits infinite number of joint-space solutions for a given task-space trajectory. However, there are various mathematical tools such as Moore-Penrose Generalized Inverse, which map the desired Cartesian trajectory into the corresponding joint-space trajectory for a kinematically redundant system. Researchers have developed various trajectory planning methods for redundant systems (Klein & Huang, 1983; Zhou & Nguyen, 1997; Siciliano, 1993; Antonelli & Chiaverini, 1998; Shi & McKay, 1986). Kinematic approach of motion planning has been reported in the past. Among them, Zhou and Nguyen (Zhou & Nguyen, 1997) formulated optimal joint-space trajectories for kinematically redundant manipulators by applying Pontryagin's Maximum Principle. Siciliano (Siciliano, 1993) has proposed an inverse kinematic approach for motion planning of redundant spacecraft-manipulator system. Antonelli and Chiaverini (Antonelli & Chiaverini, 1998) have used pseudoinverse method for task-priority redundancy resolution for an autonomous Underwater Vehicle-Manipulator System (UVMS) using a kinematic approach.

Several researchers, on the other hand, have considered dynamics of the system for trajectory planning. Among them, Vukobratovic and Kircanski (Vukobratovic & Kircanski, 1984) proposed an inverse problem solution to generate nominal joint-space trajectory considering the dynamics of the system. Bobrow (Bobrow, 1989) presented the Cartesian path of the manipulator with a B-spline polynomial and then optimized the total path traversal time satisfying the dynamic equations of motion. Shiller and Dubowsky (Shiller & Dubowsky, 1989) presented a time-optimal motion planning method considering the dynamics of the system. Shin and McKay (Shin & McKay, 1986) proposed a dynamic programming approach to minimize the cost of moving a robotic manipulator. Hirakawa and Kawamura (Hirakawa & Kawamura, 1997) have proposed a method to

solve trajectory generation problem for redundant robot manipulators using the variational approach with B-spline function to minimize the consumed electrical energy. Saramago and Steffen (Saramago & Steffen, 1998) have formulated off-line joint-space trajectories to optimize traveling time and minimize mechanical energy of the actuators using spline functions. Zhu *et al.* (Zhu *et al.*, 1999) have formulated real-time collision free trajectory by minimizing an energy function. Faiz and Agrawal (Faiz & Agrawal, 2000) have proposed a trajectory planning scheme that explicitly satisfy the dynamic equations and the inequality constraints prescribed in terms of joint variables. Recently, Macfarlane and Croft (Macfarlane & Croft, 2003) have developed and implemented a jerk-bounded trajectory for an industrial robot using concatenated quintic polynomials. Motion planning of land-based mobile robotic systems has been reported by several researchers. Among them, Brock and Khatib (Brock & Khatib, 1999) have proposed a global dynamic window approach that combines planning and real-time obstacle avoidance algorithms to generate motion for mobile robots. Huang *et al.* (Huang *et al.*, 2000) have presented a coordinated motion planning approach for a mobile manipulator considering system stability and manipulation. Yamamoto and Fukuda (Yamamoto & Fukuda, 2002) formulated trajectories considering kinematic and dynamic manipulability measures for two mobile robots carrying a common object while avoiding a collision by changing their configuration dynamically. Recently, Yamashita *et al.* (Yamashita *et al.*, 2003) have proposed a motion planning method for multiple mobile robots for cooperative transportation of a large object in a 3D environment. To reduce the computational burden, they have divided the motion planner into a global path planner and a local manipulation planner then they have designed it and integrated it. All the previously mentioned researches have performed trajectory planning for either space robotic or land-based robotic systems. On the other hand, very few works on motion/trajectory planning of underwater robotic systems have been reported so far. Among them, Yoerger and Slotine (Yoerger & Slotin, 1985) formulated a robust trajectory control approach for underwater robotic vehicles. Spangelo and Egeland (Spangelo & Egeland, 1994) developed an energy-optimum trajectory for underwater vehicles by optimizing a performance index consisting of a weighted combination of energy and time consumption by the system. Recently, Kawano and Ura (Kawano & Ura, 2002) have proposed a motion planning algorithm for nonholonomic autonomous underwater vehicle in disturbance using reinforcement learning (Q-learning) and teaching method. Sarkar and Podder (Sarkar & Podder, 2001) have presented a coordinated motion planning algorithm for a UVMS to minimize the hydrodynamic drag. Note that UVMS always implies an autonomous UVMS here.

However, majority of the trajectory planning methods available in the literature that considered the dynamics of the system are formulated for land-based robots. They have either optimized some objective functions related to trajectory planning satisfying dynamic equations or optimized energy functions. Moreover, for the land-based robotic system, the dynamics of the system is either homogeneous or very close to homogeneous. On the other hand, most of the trajectory planning methods that have been developed for space and underwater robotic systems use the pseudoinverse approach that neglects the dynamics of the system (Siciliano, 1993; Antonelli & Chiaverini, 1998; Sarkar & Podder, 2001).

In this research, we propose a new trajectory planning methodology that generates a kinematically admissible and dynamically feasible trajectory for kinematically

redundant systems whose subsystems have greatly different dynamic responses. We consider the trajectory planning of underwater robotic systems as an application to the proposed theoretical development. In general, a UVMS is composed of a 6 dof Autonomous Underwater Vehicles (AUV) and one (or more) n dof robotic manipulator(s). Commonly, the dynamic response of the AUV is an order of magnitude slower than that of the manipulator(s). Therefore, a UVMS is a kinematically redundant heterogeneous dynamic system for which the trajectory planning methods available in the literature are not directly applicable. For example, when the joint-space description of a robotic system is determined using pseudoinverse, all joints are implicitly assumed to have same or similar dynamic characteristics. Therefore, the traditional trajectory planning approaches may generate such reference trajectories that either the UVMS may not be able to track them or while tracking, it may consume exorbitant amount of energy which is extremely precious for autonomous operation in oceanic environment.

Here, we present a new unified motion planning algorithm for a UVMS, which incorporates four other independent algorithms. This algorithm considers the variability in dynamic bandwidth of the complex UVMS system and generates not only kinematically admissible but also dynamically feasible reference trajectories. Additionally, this motion planning algorithm exploits the inherent kinematic redundancy of the whole system and provides reference trajectories that accommodates other important criteria such as thruster/actuator faults and saturations, and also minimizes hydrodynamic drag. All these performance criteria are very important for autonomous underwater operation. They provide a fault-tolerant and reduced energy consuming autonomous operation framework. We have derived dynamic equations of motion for UVMS using a new approach Quasi-Lagrange formulation and also considered thruster dynamics. Effectiveness of the proposed unified motion planning algorithm has been verified by extensive computer simulation and some experiments.

2. UVMS Dynamics

The dynamics of a UVMS is highly coupled, nonlinear and time-varying. There are several methods such as the Newton-Euler method, the Lagrange method and Kane's method to derive dynamic equations of motion. The Newton-Euler approach is a recursive formulation and is less useful for controller design (Kane & Lavinson, 1985; Fu *et al.*, 1988; Craig, 1989). Kane's method is a powerful approach and it generates the equations of motion in analytical forms, which are useful for control. However, we choose to develop the dynamic model using the Lagrange approach because of two reasons. First, it is a widely known approach in other fields of robotics and thus will be accessible to a larger number of researchers. Second, this is an energy-based approach that can be easily extended to include new subsystems (e.g., inclusion of another manipulator).

There is a problem, however, to use the standard form of the Lagrange equation to derive the equations of motion of a UVMS. When the base of the manipulator is not fixed in an inertial frame, which is the case for a UVMS, it is convenient to express the Lagrangian not in terms of the velocities expressed in the inertial frame but in terms of velocities expressed in a body attached frame. Moreover, for feedback control, it is more convenient to work with velocity components about body-fixed axes, as sensors

measure motions and actuators apply torques in terms of components about the body-fixed reference frame. However, the components of the body-fixed angular velocity vector cannot be integrated to obtain actual angular displacement. As a consequence of this, we cannot use the Lagrange equation directly to derive the dynamic equations of motion in the body-fixed coordinate frame. This problem is circumvented by applying the *Quasi-Lagrange* approach. The Quasi-Lagrange approach was used earlier to derive the equations of motion of a space structure (Vukobratovic & Kircanski, 1984). Fossen mentioned the use of the same approach to model an AUV (Fossen, 1984).

However, this is the first time that a UVMS is modeled using the Quasi-Lagrange approach. This formulation is attractive because it is similar to the widely used standard Lagrange formulation, but it generates the equations of motion in the body-attached, non-inertial reference frame, which is needed in this case.

We, for convenience, commonly use two reference frames to describe underwater robotic systems. These two frames are namely the earth-fixed frame (denoted by XYZ) and the body-fixed frame (denoted by $X_v Y_v Z_v$), as shown in Fig. 1.

The dynamic equations of motion of a UVMS can be expressed as follows:

$$M_b(q_m)\dot{w} + C_b(q_m, w)w + D_b(q_m, w)w + G_b(q) = \tau_b \quad (1)$$

where the subscript 'b' denotes the corresponding parameters in the body-fixed frames of the vehicle and the manipulator. $M_b(q_m) \in \mathbb{R}^{(6+n) \times (6+n)}$ is the inertia matrix including the added mass and $C_b(q_m, w) \in \mathbb{R}^{(6+n) \times (6+n)}$ is the centrifugal and Coriolis matrix including terms due to added mass. $D_b(q_m, w) \in \mathbb{R}^{(6+n) \times (6+n)}$ is the drag matrix, $G(q) \in \mathbb{R}^{(6+n)}$ is the vector of restoring forces and $\tau_b \in \mathbb{R}^{(6+n)}$ is the vector of forces and moments acting on the UVMS. The displacement vector $q = [q_v, q_m]^T$, where $q_v = [q_1, \dots, q_6]^T$, and $q_m = [q_7, \dots, q_{6+n}]^T$. q_1, q_2 and q_3 are the linear (surge, sway, and heave) displacements of the vehicle along X, Y, and Z axes, respectively, expressed in the earth-fixed frame. q_4, q_5 and q_6 are the angular (roll, pitch, and yaw) displacements of the vehicle about X, Y and Z axes, respectively, expressed in the earth-fixed frame. q_7, q_8, \dots, q_{6+n} are the angular displacements of joint 1, joint 2, ..., joint n of the manipulator in link-fixed frames. The quasi velocity vector $w = [w_1, \dots, w_{6+n}]^T$, where w_1, w_2 and w_3 are the linear velocities of the vehicle along X_v, Y_v , and Z_v axes respectively, expressed in the body-fixed frame. w_4, w_5 and w_6 are the angular velocities of the vehicle about X_v, Y_v , and Z_v axes, respectively, expressed in the body-fixed frame. w_7, w_8, \dots, w_{6+n} are the angular velocities of manipulator joint 1, joint 2, ..., joint n , expressed in the link-fixed frame. A detailed derivation of Equation (1) is given in (Podder, 2000).

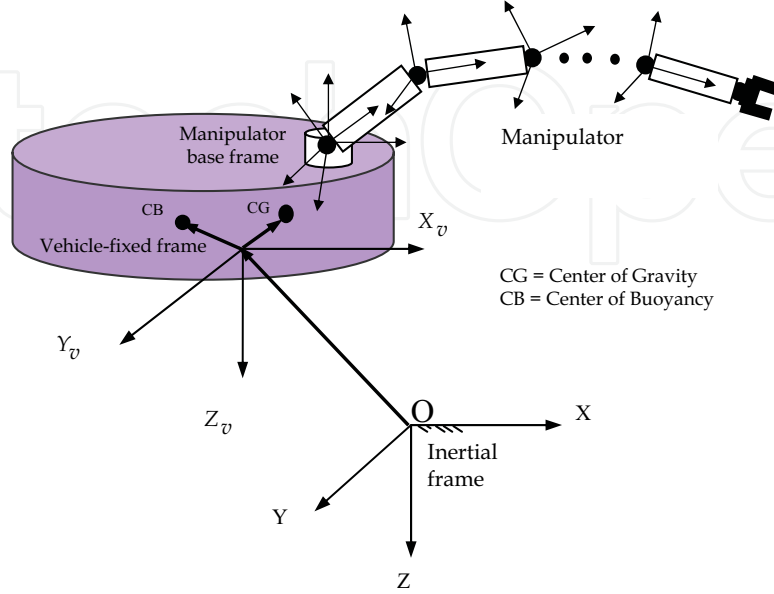


Fig. 1. Coordinate frames for underwater vehicle-manipulator system.

Equation (1) is represented in the body-fixed frame of the UVMS because it is convenient to measure and control the motion of the UVMS with respect to the moving frame. However, the integration of the angular velocity vector does not lead to the generalized coordinates denoting the orientation of the UVMS. In general, we can relate the derivative of the generalized coordinates and the velocity vector in the body-fixed frame by the following linear transformation:

$$\dot{q} = Bw \quad (2)$$

The transformation matrix B in Equation (2) is given by:

$$B(q) = \begin{bmatrix} B_{1 \times 6} & O_{6 \times n} \\ O_{n \times 6} & B_2 \end{bmatrix}_{(6+n) \times (6+n)}, \quad B_1 = \begin{bmatrix} J_1 & O \\ O & J_2 \end{bmatrix}, \quad B_2 = [I] \quad (3)$$

where the linear velocity transformation matrix, J_1 , and the angular velocity transformation matrix, J_2 , are given as:

$$J_1 = \begin{bmatrix} C_5 C_6 & -S_6 C_4 + S_4 S_5 C_6 & S_4 S_6 + S_5 C_4 C_6 \\ S_6 C_5 & C_4 C_6 + S_4 S_5 S_6 & -S_4 C_6 + S_5 S_6 C_4 \\ -S_5 & S_4 C_5 & C_4 C_5 \end{bmatrix} \quad (4)$$

$$J_2 = \begin{bmatrix} 1 & S_4 T_5 & C_4 T_5 \\ 0 & C_4 & -S_4 \\ 0 & S_4 / C_5 & C_4 / C_5 \end{bmatrix} \quad (5)$$

Here S_i , C_i and T_i represent $\sin(q_i)$, $\cos(q_i)$ and $\tan(q_i)$, respectively, and I is the identity matrix. Note that there is an Euler angle (roll, pitch, yaw) singularity in J_2 when the pitch angle (q_5) is an odd multiple of $\pm 90^\circ$. Generally, the pitch angle in practical operation is

restricted to $|q_5| < 90^\circ$. However, if we need to avoid singularity altogether, unit quaternions can be used to represent orientation (Fossen, 1984).

3. Dynamics-Based Trajectory Planning Algorithm

Most of the trajectory planning methods found in literature is formulated for land-based robots where the dynamics of the system is homogeneous or very close to homogeneous. The study of UVMS becomes more complicated because of the heterogeneous dynamics and dynamic coupling between two different bandwidth subsystems. From practical point of view it is very difficult and expensive to move a heavy and large body with higher frequency as compared to a lighter and smaller body. The situation becomes worse in the case of underwater systems because of the presence of heavier liquid (water) which contributes significant amount of drag forces. Therefore, it will be more meaningful if we can divide the task into several segments depending on the natural frequencies of the subsystems. This will enable the heterogeneous dynamic system to execute the trajectory not only kinematically admissibly but also dynamically feasibly.

Here we present a trajectory planning algorithm that accounts for different bandwidth characteristic of a dynamic system. First, we present the algorithm for a general n -bandwidth dynamic system. Then we improvise this algorithm for application to a UVMS.

3.1 Theoretical Development

Let us assume that we know the natural frequency of each subsystem of the heterogeneous dynamic system. This will give us a measure of the dynamic response of each subsystem. Let these frequencies be $\omega_i, i = 1, 2, \dots, s$.

We approximate the task-space trajectories using Fourier series and represent it in terms of the summation of several frequencies in ascending order.

$$x_{d \times 1}(t) = f_{6 \times 1}(t) = a_0 + \sum_{r=1}^{\infty} a_r \cos(r\pi t / L) + \sum_{r=1}^{\infty} b_r \sin(r\pi t / L) \quad (6)$$

where a_0, a_r, b_r are the coefficients of Fourier series and are represented as 6×1 column vectors, $r/2L$ is the frequency of the series and $2L$ is the time period.

Now we truncate the series at a certain value of r (assuming $r = p_1$ to be sufficiently large) so that it can represent the task-space trajectories reasonably. We rewrite the task-space trajectory in the following form:

$$x_{d \times 1}(t) = f_{6 \times 1}(t) = f_1(t) + f_2(t) + \dots + f_{p_1}(t) \quad (7)$$

where $f_1(t) = a_0 + a_1 \cos(\pi t / L) + b_1 \sin(\pi t / L)$, and $f_j(t) = a_j \cos(j\pi t / L) + b_j \sin(j\pi t / L)$ for $j = 2, 3, \dots, p_1$.

We then use these truncated series as the reference task-space trajectories and map them into the desired (reference) joint-space trajectories by using weighted pseudoinverse method as follows:

$$\dot{q}_{d_j} = J_{W_j}^+ \dot{x}_{d_j} \quad (8)$$

$$\ddot{q}_{d_j} = J_{W_j}^+ (\ddot{x}_{d_j} - \dot{J} \dot{q}_{d_j}) \quad (9)$$

where \dot{q}_{d_j} are the joint-space velocities and \ddot{q}_{d_j} are the joint-space accelerations corresponding to the task-space velocities $\dot{x}_{d_j} = d(f_j(t))/dt$ and task-space accelerations $\ddot{x}_{d_j} = d^2(f_j(t))/dt^2$ for $j = 1, 2, \dots, p_1$. $J_{w_j}^+ = W_j^{-1} J^T (J W_j^{-1} J^T)^{-1}$ are the weighted pseudoinverse of Jacobians and $W_j = \text{diag}(h_{1_j}, \dots, h_{(6+n)_j})$ are diagonal weight matrices.

In our proposed scheme we use weighted pseudoinverse technique in such a way that it can act as a filter to remove the propagation of undesirable frequency components from the task-space trajectories to the corresponding joint-space trajectories for a particular subsystem. This we do by putting suitable zeros in the diagonal entries of the W_j^{-1} matrices in Equation (8) and Equation (9). We leave the other elements of W_j^{-1} as unity. We have developed two cases for such a frequency-wise decomposition as follows:

Case I – Partial Decomposition:

In this case, the segments of the task-space trajectories having frequencies ω_t ($\omega_t \leq \omega_i$) will be allocated to all subsystems that have natural frequencies greater than ω_t up to the maximum bandwidth subsystem. To give an example, for a UVMS, the lower frequencies will be shared by both the AUV and the manipulator, whereas the higher frequencies will be solely taken care of by the manipulator.

Case II- Total Decomposition:

In this case, we partition the total system into several frequency domains, starting from the low frequency subsystem to the very high frequency subsystem. We then allocate a particular frequency component of the task-space trajectories to only those subsystems that belong to the frequency domain just higher than the task-space component to generate joint-space trajectories. For a UVMS, this means that the lower frequencies will be taken care of by the vehicle alone and the higher frequencies by the manipulator alone.

To improvise the general algorithm for a $(6+n)$ dof UVMS, we decompose the task-space trajectories into two components as follows:

$$f(t) = f_{11}(t) + f_{22}(t) \quad (10)$$

where $f_{11}(t) = a_0 + \sum_{r=1}^{r_1} a_r \cos(r\pi t/L) + \sum_{r=1}^{r_1} b_r \sin(r\pi t/L)$, $f_{22}(t) = \sum_{r=r_1+1}^{r_2} a_r \cos(r\pi t/L) + \sum_{r=r_1+1}^{r_2} b_r \sin(r\pi t/L)$, r_1 and r_2 ($r_2 = p_1$) are suitable finite positive integers. Here, $f_{11}(t)$ consists of lower frequency terms and $f_{22}(t)$ has the higher frequency terms.

Now, the mapping between the task-space variables and the joint-space variables are performed as

$$\ddot{q}_{d_1} = J_{W_1}^+ (\ddot{x}_{d_1} - \dot{J} \dot{x}_{d_1}) \quad (11)$$

$$\ddot{q}_{d_2} = J_{W_2}^+ (\ddot{x}_{d_2} - \dot{J} \dot{x}_{d_2}) \quad (12)$$

$$\ddot{q}_d = \ddot{q}_{d_1} + \ddot{q}_{d_2} \quad (13)$$

where $W_i \in \mathbb{R}^{(6+n) \times (6+n)}$ are the weight matrices, $\ddot{q}_d \in \mathbb{R}^{(6+n)}$ are the joint-space accelerations and $J_{wi}^+ = W_i^{-1} J^T (J W_i^{-1} J^T)^{-1}$ for $(i=1,2)$. We have considered the weight matrices for two types of decompositions as follows:

For Case I – Partial decomposition:

$$W_1 = \text{diag}(h_1, h_2, \dots, h_{6+n}) \quad (14)$$

$$W_2 = \text{diag}(0, \dots, 0, h_7, \dots, h_{6+n}) \quad (15)$$

For Case II- Total decomposition:

$$W_1 = \text{diag}(h_1, \dots, h_6, 0, \dots, 0) \quad (16)$$

$$W_2 = \text{diag}(0, \dots, 0, h_7, \dots, h_{6+n}) \quad (17)$$

The weight design is further improved by incorporating the system's damping into the trajectory generation for UVMS. A significant amount of energy is consumed by the damping in the underwater environment. Hydrodynamic drag is one of the main components of such damping. Thus, if we decompose the motion in the joint-space in such a way that it is allocated in an inverse ratio to some measure of damping, the resultant trajectory is expected to consume less energy while tracking the same task-space trajectory. Thus, we incorporate the damping into the trajectory generation by designing the diagonal elements of the weight matrix as $h_i = f(\zeta_i)$, where ζ_i ($i=1, \dots, 6+n$) is the damping ratio of the particular dynamic subsystem which can be found out using multi-body vibration analysis techniques (James *et al.*, 1989). A block diagram of the proposed scheme has been shown in Fig. 2.

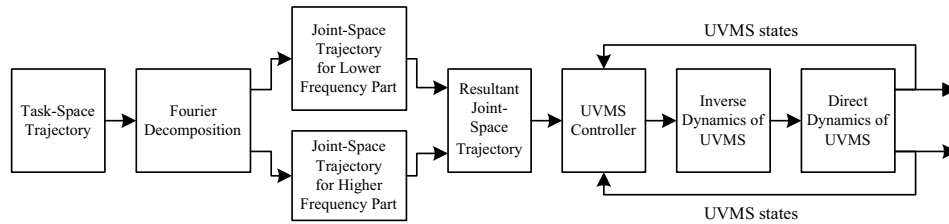


Fig. 2. Dynamics-based planning scheme.

3.2 Implementation Issues

It is to be noted that in the proposed dynamics-based method we have decomposed the task-space trajectory into two domains where the lower frequency segments of the task-space trajectories are directed to either the heavier subsystem, i.e., the vehicle in Case II, or to both the heavier and lighter subsystems, i.e., the vehicle and the manipulator as in Case I. The high frequency segments of the task-space trajectories, on the other hand, are always allocated to the lighter subsystem, i.e., the manipulator. These allocations of task-space trajectories have been mapped to corresponding joint-space trajectories by utilizing weighted pseudoinverse technique where the heterogeneous dynamics of the UVMS have been taken into consideration. Then, these reference joint-space trajectories are followed by the individual joint/ dof to execute the end-effector's trajectories.

There are two basic issues of this proposed algorithm that must be discussed before it can be implemented. They are: given a nonlinear, multi degree-of-freedom (n -DOF) dynamic

system having different frequency bandwidth subsystems, how to find the 1) natural frequencies of each subsystem, and 2) the damping ratios of each subsystem. We briefly point out the required steps that are needed to obtain these system dynamic parameters: (1) Linearize the dynamic equations, (2) Find the eigenvalues and eigenvectors from the undamped homogeneous equations, (3) Find the orthogonal modal matrix (P), (4) Find the generalized mass matrix ($P^T MP$), (5) Find the generalized stiffness matrix ($P^T KP$), (6) Find the weighted modal matrix (\tilde{P}), (7) Using Rayleigh damping equation find a proportional damping matrix, and (8) Decouple the dynamic equations by using \tilde{P} .

After all these operations, we will obtain $(6+n)$ decoupled equations similar to that of a single-dof system instead of $(6+n)$ coupled equations. From this point on, finding the natural frequencies (ω_i) and the damping ratios (ζ_i) are straightforward. A detailed discussion on these steps can be found in advanced vibration textbook (James *et al.*, 1989).

3.3 Results and Discussion

We have conducted extensive computer simulations to investigate the performance of the proposed Drag Minimization (DM) algorithm. The UVMS used for the simulation consists of a 6 dof vehicle and a 3 dof planar manipulator working in the vertical plane. The vehicle is ellipsoidal in shape with length, width and height $2.0m$, $1.0m$ and $1.0m$, respectively. The mass of the vehicle is $1073.0Kg$. The links are cylindrical and each link is $1.0m$ long. The radii of link 1, 2 and 3 are $0.1m$, $0.08m$ and $0.07m$, respectively. The link masses (oil filled) are $32.0Kg$, $21.0Kg$ and $16.0Kg$, respectively. We have compared our results with that of the conventional Pseudoinverse (PI) method (i.e., without the null-space term), which is a standard method for resolving kinematic redundancy.

3.3.1 Trajectory

We have chosen a square path in xy (horizontal) plane for the computer simulation. We have assumed that each side of the square path is tracked in equal time. The geometric path and the task-space trajectories are given in Fig. 3.

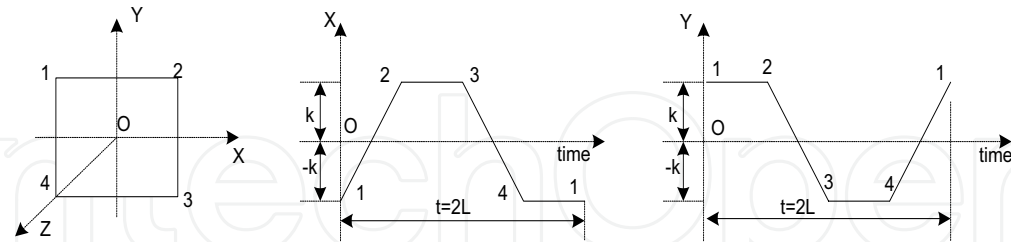


Fig. 3. Task-space geometric path and trajectories.

The task-space trajectories can be represented as

$$x(t) = f_x(t) = \begin{cases} 4kt/L - k & \text{if } 0 < t \leq L/2 \\ k & \text{if } L/2 < t \leq L \\ 5k - 4kt/L & \text{if } L < t \leq 3L/2 \\ -k & \text{if } 3L/2 < t \leq 2L \end{cases} \quad (18)$$

$$y(t) = f_y(t) = \begin{cases} k & \text{if } 0 < t \leq L/2 \\ 3k - 4kt/L & \text{if } L/2 < t \leq L \\ -k & \text{if } L < t \leq 3L/2 \\ 4kt/L - 7k & \text{if } 3L/2 < t \leq 2L \end{cases} \quad (19)$$

$$z(t) = f_z(t) = 0 \quad (20)$$

The Fourier series for the above trajectories are as follows:

$$f_j(t) = a_{0j} + \sum_{r=1}^{\infty} a_{rj} \cos(r\pi t/L) + \sum_{r=1}^{\infty} b_{rj} \sin(r\pi t/L) \quad (21)$$

where 'j' implies the coefficients for x , y or z ; k is a constant and $2L$ is the time period. The Fourier coefficients are:

$$a_{0x} = -a_{0y} = k, \quad a_{rx} = -a_{ry} = 4k/(r\pi)^2 (\cos r\pi - 1) \quad \text{and} \quad b_{rx} = -b_{ry} = 8k/(r\pi)^2 \sin(r\pi/2).$$

For this simulation, we have taken $k = 1m$, i.e., the path is $2m$ square, $L=5$ and maximum frequency at which the Fourier series is truncated is $r = p_1 = 30$. The frequency of the manipulator is 10 times higher than that of the vehicle. We have taken the natural frequency of the vehicle as 0.15 cycles per second and the manipulator to be 10 time faster than the vehicle. We have segmented the task-space trajectories as

$$f_{j1}(t) = a_{0j} + a_{1j} \cos(\pi t/L) + b_{1j} \sin(\pi t/L) \quad (22)$$

$$f_{j2}(t) = \sum_{r=2}^{30} a_{rj} \cos(r\pi t/L) + \sum_{r=2}^{30} b_{rj} \sin(r\pi t/L) \quad (23)$$

We have compared our results from the proposed dynamics-based trajectory planning method with that from the conventional straight-line trajectory planning method using regular pseudoinverse technique. In conventional method, the trajectory is designed in three sections: the main section (intermediate section), which is a straight line, is preceded and followed by two short parabolic sections (Fu *et al.*, 1988; Craig, 1989). The simulation time is 10.0sec, which is required to complete the square path in XY (horizontal) plane. The total length of the path is 8.0m; the average speed is about 1.6knot. This speed is more than JASON vehicle (speed = 1.0knot) but less than SAUVIM system (designed speed = 3.0knot).

We have presented results from computer simulations in Fig. 4 through Fig. 9. Results for Case I (Partial Decomposition) are plotted in Fig. 5 through Fig. 7 and that of for Case II (Total Decomposition) are provided in Fig. 8 through Fig. 9. It is observed from Fig. 4 and 5 that the end-effector tracks the task-space paths and trajectories quite accurately. The errors are very small. The joint-space trajectories are plotted in Fig. 6. It is observed that the proposed dynamics-based method restricts the motion of the heavy subsystem and allows greater motion of the lighter subsystem to track the trajectory. It is also noticed that the motion of the heavy subsystem is smoother. The errors in joint-space trajectory are almost zero.

Simulation results for surge-sway motion, power requirement and energy consumption for conventional straight-line method are plotted in the left column and that of for proposed dynamics-based method are plotted in the right column in Fig. 7. Top two plots of Fig. 7 show the differences in surge-sway movements for two methods. In case of the conventional method, the vehicle changes the motion very sharply as compared to the motion generated from the dynamics-based method. It may so happen that this type of sharp movements may be beyond the capability of the heavy dynamic subsystem and consequently large errors in trajectory tracking may occur. Moreover, the vehicle will experience large velocity and

acceleration in conventional method that result in higher power requirement and energy consumption, as we observe in Fig. 7.

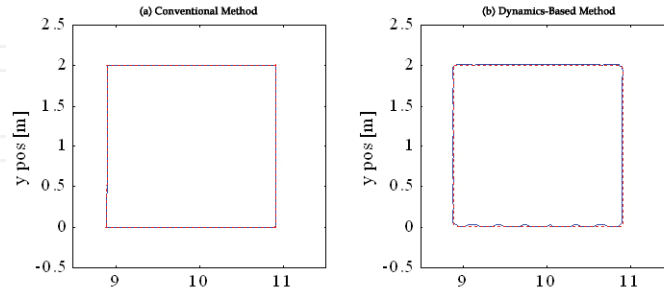


Fig. 4. Task-space geometric paths, (a) Conventional Straight-line planning method and (b) Dynamics-Based planning method for Case I. The actual path is denoted by solid line and the desired path is denoted by dashed line.

We have also presented simulation results for Case II (Total Decomposition) in Fig. 8 and 9. From Fig. 8 it is observed that even though the vehicle has moved more as compared to the conventional straight-line planning method, the motion is smooth. This type of motion is more realistic for a heavy subsystem like the vehicle here and it also avoids large acceleration of the vehicle. On the other hand, the movement of the manipulator is smaller but sharper than that of the conventional method. In the plots in the left column of Fig. 9 it is shown that the end-effector tracks the task-space trajectories quite accurately. The second plot on the right column of this figure shows that the power requirement of the UVMS is less in Case II of the proposed dynamics-based method as compared to that of in conventional method.

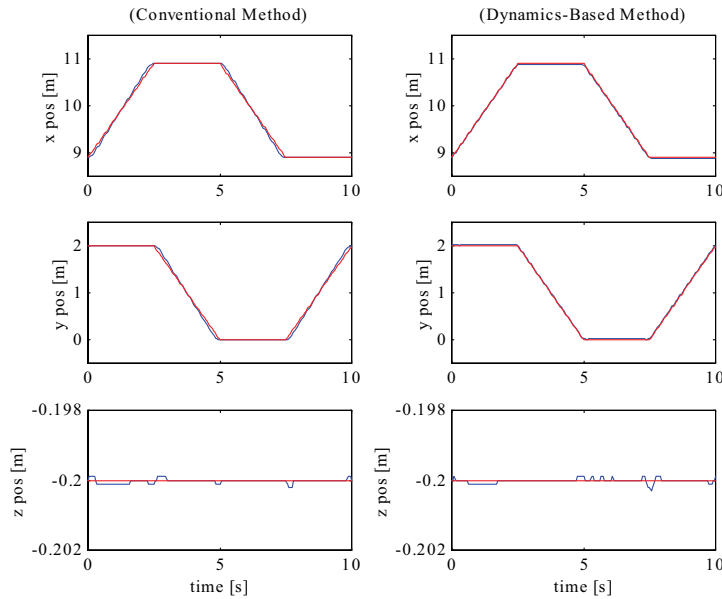


Fig. 5. Task-space trajectories: Conventional Straight-line planning method (left column) and Dynamics-Based planning method for Case I (right column). Desired trajectories are denoted by dashed lines and actual trajectories are denoted by solid lines.

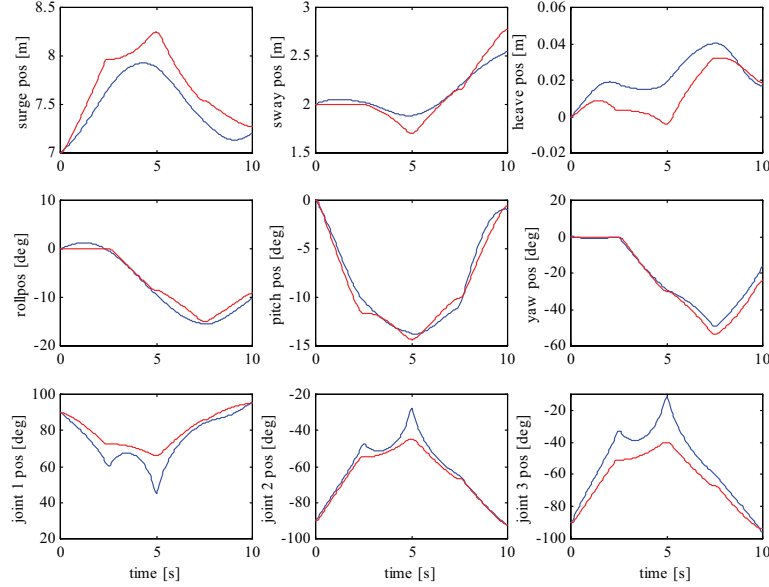


Fig. 6. Joint-space trajectories: Dynamics-Based planning method for Case I (solid/blue line) and Conventional Straight-line planning method (dashed/red line).

For Case II, we can say even though the reduction of energy consumption is not much, however, the movement is smooth that can be practically executed. The power requirement is also less as compared to the conventional straight-line planning method.

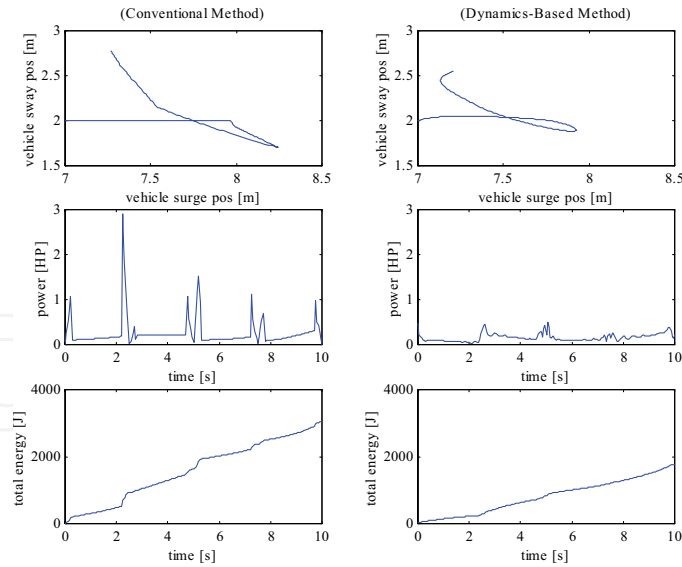


Fig. 7. X-Y motion of the center of gravity, power and energy consumption of the UVMS. Left column for Conventional Straight-line planning method and right column for Dynamics-Based planning method for Case I (partial decomposition).

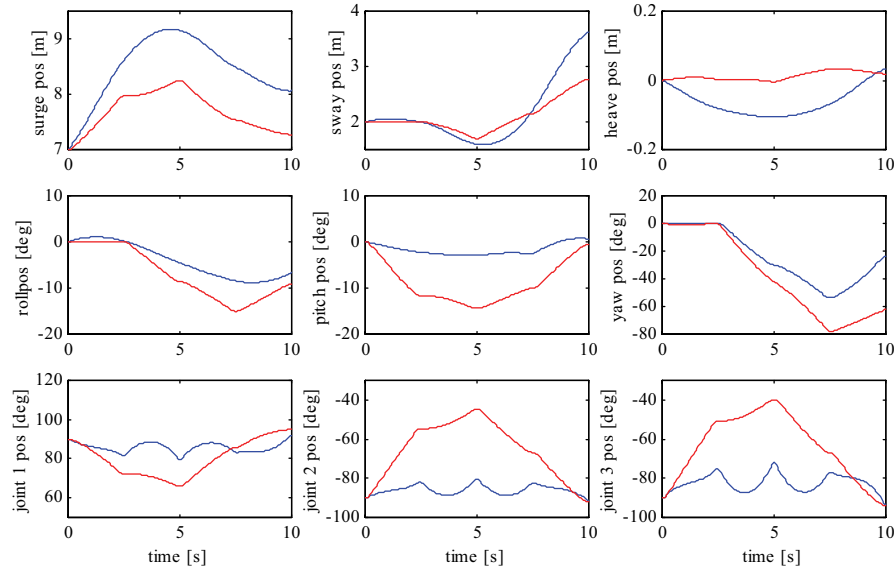


Fig. 8. Joint-space trajectories: Dynamics-Based planning method (solid/blue line) for Case II and Conventional Straight-line planning method (dashed/red line).

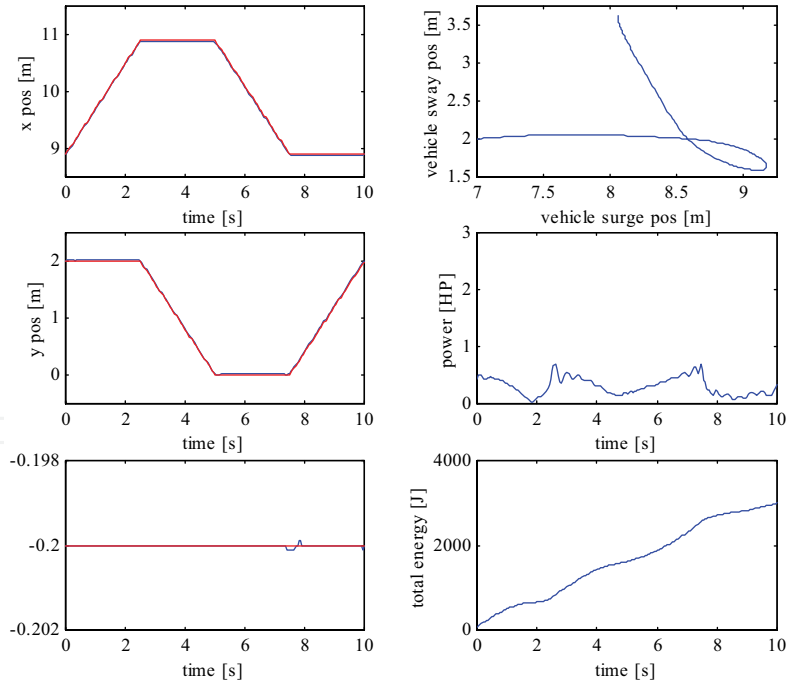


Fig. 9. Task-space trajectories (left column), and surge-sway motion, power requirement and energy consumption (right column) for Dynamics-Based planning method for Case II (total decomposition).

4. Fault Tolerant Decomposition Algorithm

A UVMS is expected to function in a hazardous and unstructured underwater environment. A thruster/actuator fault can occur due to various reasons. There are different methods to detect and isolate these faults. Without going into the details of the possible nature of thruster/actuator faults and how they can be detected and isolated, we assume in this work that we can detect and isolate thruster/actuator faults when they occur. In general, there are more thrusters and actuators than what is minimally required for the specific dof that a UVMS is designed for. Here, we develop an algorithm to exploit the thruster and actuator redundancy to accommodate thruster/actuator faults during operation.

4.1 Theoretical Development

In order to relate the generalized force vector τ_b with the individual thruster/actuator force/torque, let us consider a UVMS which has p thrusters and actuators where, in general, $p \geq (6+n)$. In such a case, we can write

$$\tau_b = EF_t \quad (24)$$

where $E \in \mathbb{R}^{(6+n) \times p}$ thruster configuration matrix and $F_t \in \mathbb{R}^p$ is the reference thruster and actuator forces and torques. The thruster configuration matrix is a constant matrix that depends on the geometric locations of the thrusters and actuators.

Substituting Equation (24) into Equation (1) and performing algebraic manipulation we get

$$\dot{w} = M_b^{-1}(EF_{td} - \xi_b) \quad (25)$$

where $\xi_b = C_b(q_m, w)w + D_b(q_m, w)w + G_b(q)$.

Differentiation of Equation (2) leads to the following acceleration relationship:

$$\ddot{q} = B\dot{w} + \dot{B}w \quad (26)$$

Now, from Equation (25) and Equation (26) we can write

$$\ddot{q} = \eta F_t + \dot{\lambda} \quad (27)$$

where $\eta_{(6+n) \times p} = BM_b^{-1}E$ and $\dot{\lambda}_{(6+n) \times 1} = \dot{B}w - BM_b^{-1}\xi_b$.

From Equation (27), using weighted pseudoinverse technique we obtain a least-norm solution to thruster and actuator forces and torques as

$$F_t = \eta_W^+ (\ddot{q} - \dot{\lambda}) \quad (28)$$

where $\eta_W^+ = W^{-1} \eta^T (\eta W^{-1} \eta^T)^{-1}$ is the weighted pseudoinverse of η and $W = \text{diag}(h_1, h_2, \dots, h_p)$ is the weight matrix.

Now, we construct a thruster fault matrix, $\psi_{p \times p} = W^{-1}$, with diagonal entries either 1 or 0 to capture the fault information of each individual thruster/actuator. If there is any thruster/actuator fault we introduce 0 into the corresponding diagonal element of ψ , otherwise it will be 1. We can also rewrite Equation (28) in terms of thruster fault matrix, ψ , as

$$F_t = \psi \eta^T (\eta \psi \eta^T)^{-1} (\ddot{q} - \dot{\lambda}) \quad (29)$$

Equation (29) provides us the fault tolerant allocation of thruster/actuator force/torque, F_t . More detailed discussion on this topic can be found in (Podder & Sarkar, 2000; Podder *et al.*, 2001).

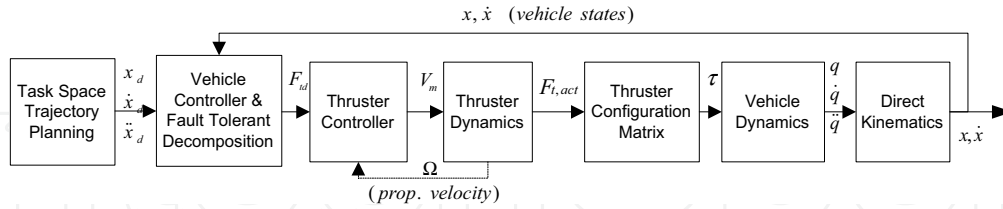


Fig. 10. Fault-tolerant control scheme.

4.2 Experimental Setup

We have conducted both computer simulations and underwater experiments to verify the proposed fault-tolerant control scheme. We have used ODIN (Omni-Directional Intelligent Navigator), which is a 6 dof vehicle designed at the University of Hawaii], as our test-bed. ODIN is a near-spherical AUV that has 4 horizontal thrusters and 4 vertical thrusters as shown in Fig. 11. We have compared our simulation results with that of actual experiments, and presented them later in this section.

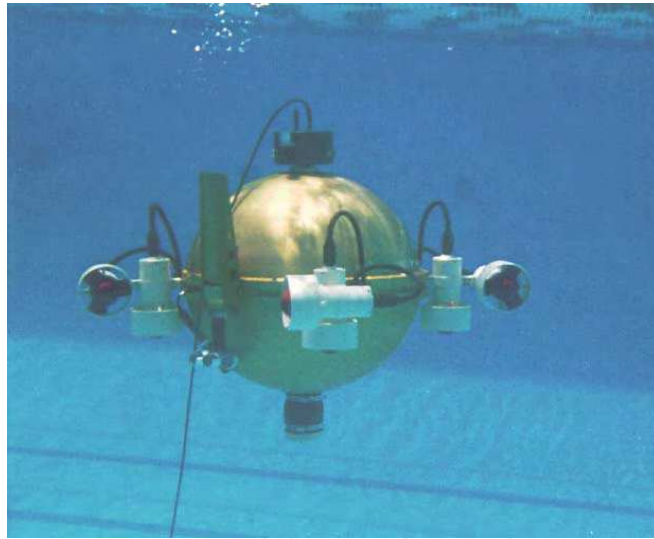


Fig. 11. Omni-Directional Intelligent Navigator (ODIN) vehicle.

The ODIN has a near-spherical shape with horizontal diameter of $0.63m$ and vertical diameter of $0.61m$, made of anodized Aluminum (AL 6061-T6). Its dry weight is $125.0Kg$ and is slightly positively buoyant. The processor is a Motorola 68040/33MHz working with VxWorks 5.2 operating systems. The RS232 protocol is used for RF communication. The RF Modem has operating range up to $480m$, operating frequency range $802-928MHz$, and maximum transmission speed $38,400$ baud data rate. The power supply is furnished by 24 Lead Gel batteries, where 20 batteries are used for the thrusters and 4 batteries are used for the CPU. ODIN can perform two hours of autonomous operation.

The actuating system is made of 8 thrusters of which 4 are vertical and 4 are horizontal. Each thruster has a brushless DC motor weighing approximately $1Kg$ and can provide a maximum thrust of approximately $27N$. The sensor system is composed of: 1) a pressure

sensor for measuring depth with an accuracy of 3cm , 2) 8 sonars for position reconstruction and navigation, each with a range of $0.1\text{--}14.4\text{m}$, and 3) an inertial system for attitude and velocity measurement. Since the sonars need to be in the water to work properly, the first 100sec of sonar data is not accurate.

The experiments were conducted at the University of Hawaii swimming pool. Several experiments were performed to verify the proposed control scheme. The thruster faults were simulated by imposing zero voltages to the relevant thrusters.

4.3 Results and Discussion

We have performed extensive computer simulations and a number of experiments to verify the proposed planning and control scheme. We present simulation results for two cases to demonstrate the effectiveness of the proposed method. In Case 1, all thrusters are in working condition and therefore the thruster fault matrix Ψ becomes an identity matrix. In Case 2, there are two thrusters that stop working during trajectory tracking operation. In both the cases, ODIN tries to track the following trajectories: it first moves toward the z-direction for 120sec to reach a depth of 2m . Then it moves toward the y-direction for another 120.0sec to traverse 2.0m . It subsequently moves towards the x-direction for 120sec to traverse 2m . Finally it hovers at that position for another 40sec . ODIN follows a trapezoidal velocity profile during this task. The attitudes are always kept constant at $[0^\circ 0^\circ 90^\circ]$. For Case 2, one horizontal thruster (Thruster 6) fails at 260sec and one vertical thruster (Thruster 2) fails at 300sec while tracking the same trajectories as explained in Case 1. In simulations, we have introduced sensory noise in position and orientation measurements. We have chosen Gaussian noise of 2mm mean and 1.5mm standard deviation for the surge, sway and heave position measurements, 0.15degree mean and 0.15degree standard deviation for the roll, pitch and yaw position measurements for the vehicle.

In Fig. 12, we present results from a trajectory following task when there is no thruster fault. It can be observed that both the simulation and the experimental results for all the six trajectories match their respective desired trajectories within reasonable limits. It should also be noted that the particular sonar system of ODIN requires 100.0sec before it works properly. Thus, x and y trajectories in experiments have data after 100.0sec . However, the depth and attitude sensors provide information from the beginning of the task. In Fig. 13, the same trajectory following task is performed but with thruster faults. In this case, one horizontal thruster (Thruster 6) fails at 260.0sec (marked as 'A') and one vertical thruster (Thruster 2) fails at 300.0sec (marked as 'B'). Both the faulty thrusters are located at the same thruster bracket of the ODIN. Thus, this situation is one of the worst fault conditions. The simulation results are not affected by the occurrence of faults except in the case of the yaw trajectory, which produces a small error at the last part of the trajectory. In experiment, the first fault does not cause any tracking error. There are some small perturbations after the second fault from which the controller quickly recovers. It can also be noticed that in case of experiment the tracking performance is better in z-direction (depth) as compared to other two directions, i.e., x-direction and y-direction. This happened because of two reasons: 1) less environmental and hydrodynamic disturbances in z-direction, and 2) the pressure sensor for depth measurement is more accurate as compared to sonar sensors used to measure x-position and y-position. However, the orientation of the AUV, which is measured by INS sensors, is reasonably good.

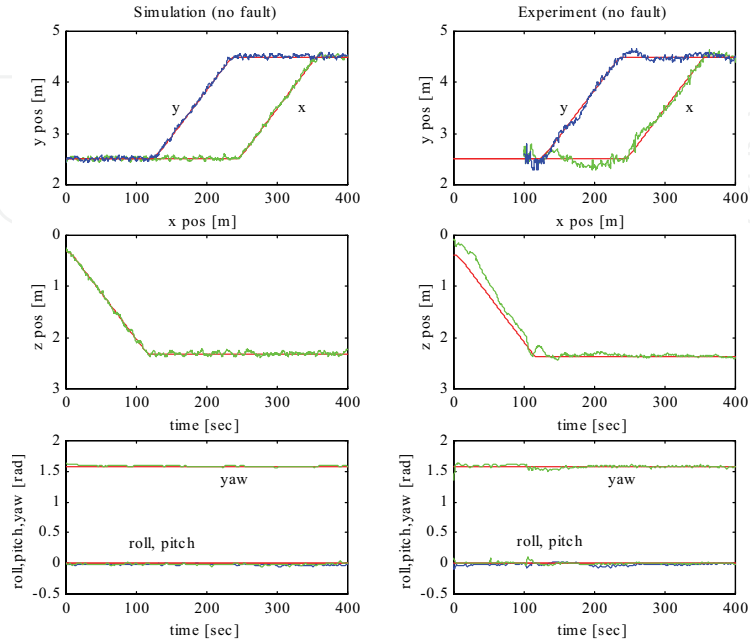


Fig. 12. Trajectory tracking with no thruster fault, simulation results in the left and experimental results in the right. The actual trajectories are denoted by solid lines and the desired trajectories are denoted by dashed lines.

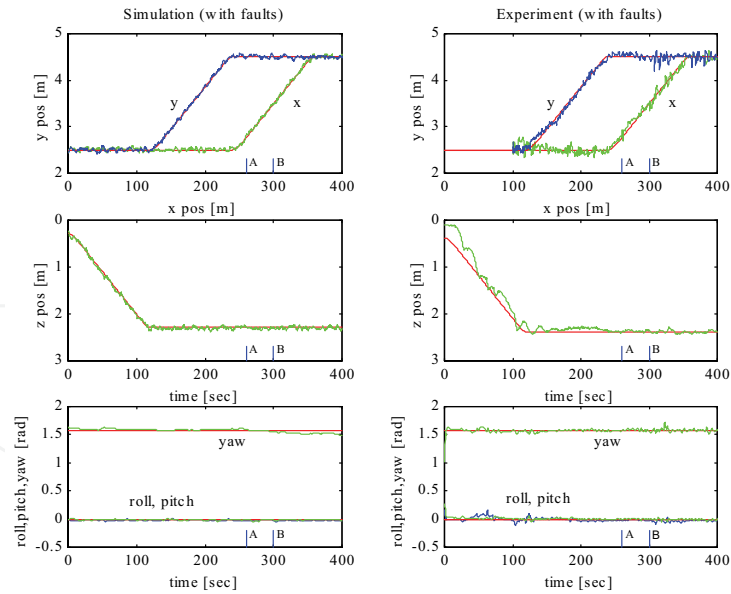


Fig. 13. Trajectory tracking with thruster faults, simulation results in the left and experimental results in the right. The actual trajectories are denoted by solid lines and the desired trajectories are denoted by dashed lines.

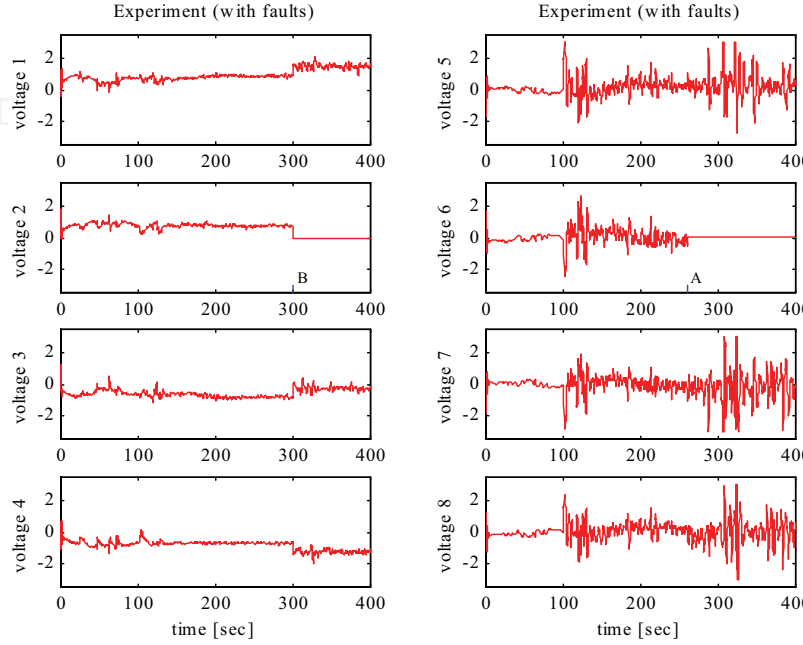


Fig. 14. Voltage versus time plots for the vertical thrusters (left) and the horizontal thrusters (right).

The voltage plots for Case 2 are presented in Fig. 14. It can be seen that voltage for Thruster 6 is zero after 260sec and that of Thruster 2 after 300sec, which imply thruster faults. From these plots it is observed that in case of simulations all the thruster voltages and in case of experiment the vertical thruster voltages are within $\pm 2\text{volt}$. Whereas, the horizontal thruster voltages in case of experiment have some spikes greater than $\pm 2\text{volt}$. The causes are as mentioned previously. We also observe that the voltage profile for the vertical thrusters matches well between simulations and experiments. This match was less obvious for horizontal thrusters. However, in all the cases, the range and general pattern seem to be consistent. More details can be found in (Podder *et al.*, 2001).

5. Saturation Limit Algorithm

In the previous section, we have derived Equation (29) for desired thruster/actuator force/torque allocation that allows the operation of the UVMS with faults. However, it cannot guarantee that the desired allocated forces/torques will remain within the saturation limit of the thrusters/actuators. As a result, if some of the forces and torques determined from those equations are beyond the capacity of the corresponding thrusters and actuators, the performance of the controller will suffer because of saturation effect.

5.1 Theoretical Development

The saturation problem cannot be solved based on the formulation given by Equations (29). In order to avoid the saturation effect, the thruster/actuator force/torque must be

controlled so that it cannot reach the saturation limit. However, in such a case, since the input to the controller and the output of the controller will be algebraically related, static state feedback technique will not be able to control the thruster and actuator forces and torques. We, therefore, propose to use the dynamic state feedback technique (Isidori *et al.*, 1968; Yun, 1988) to generate thruster forces that are within the saturation limit. The basic idea of dynamic state feedback is to introduce integrators at the input channel to enlarge the state space, and then apply the static state feedback on the enlarged system.

However, there is a difference between explicit control of the thruster/actuator forces/torques so that it can be regulated about a point or can follow a trajectory, and keep it within a specified range without regulation. The former is a force control problem and the latter is a saturation problem. We first use the dynamic state feedback technique to enlarge the state space in the following way.

We differentiate the output Equation (27) to obtain a new input V as follows:

$$\ddot{q} = \eta \dot{F}_t + \dot{\eta} F_t + \dot{\lambda} = \eta V + \gamma \quad (30)$$

where $\gamma = \eta F_t + \dot{\lambda}$ and $V = \dot{F}_t$.

Now, we consider the following control law

$$V = \eta_{WF}^* [\ddot{q}_d + K_1(\ddot{q}_d - \ddot{q}) + K_2(\dot{q}_d - \dot{q}) + K_3(q_d - q)] - \gamma \quad (31)$$

and integration of Equation (31) yields the desired thruster and actuator forces and torques as

$$F_{td} = \int V dt \quad (32)$$

where $\eta_{WF}^* = W_F^{-1} \eta^T (\eta W_F^{-1} \eta^T)^{-1}$, K_1 is the acceleration gain, K_2 is the velocity gain and K_3 is the position gain; q_d and F_{td} are desired parameters of q and F_t , respectively. The diagonal elements of the weight matrix, $W_F = \text{diag}(\hbar_1, \hbar_2, \dots, \hbar_p)$, are computed from the thruster/actuator saturation limits as follows:

We define a function of thruster and actuator force and torque variables as

$$H(F_t) = \sum_{i=1}^p \frac{1}{C_i} \frac{(F_{t,i,\max} - F_{t,i,\min})}{(F_{t,i,\max} - F_{t,i})(F_{t,i} - F_{t,i,\min})} \quad (33)$$

where $F_{t,i,\max}$ and $F_{t,i,\min}$ are the upper and lower limits of thrust/torque of the i -th thruster/actuator, and C_i is a positive quantity which is determined from the damping property of the dynamic system. Then, differentiating Equation (24), we obtain

$$\frac{\partial H(F_t)}{\partial F_{t,i}} = \frac{(F_{t,i,\max} - F_{t,i,\min})^2 (2F_{t,i} - F_{t,i,\max} - F_{t,i,\min})}{C_i (F_{t,i,\max} - F_{t,i})^2 (F_{t,i} - F_{t,i,\min})^2} \quad (34)$$

Then, the diagonal elements of the weight matrix are defined as

$$\hbar_i = 1 + |\partial H(F_t) / \partial F_{t,i}| \quad (35)$$

From the above expression (34), we notice that $\partial H(F_t) / \partial F_{t,i}$ is equal to zero when the i -th thruster/actuator is at the middle of its range, and becomes infinity at either limits. Thus, \hbar_i varies from 1 to infinity if the i -th thrust goes from middle of the range to its limit. If the i -th thrust/torque approaches its limit, then \hbar_i becomes very large and the corresponding element in W_F^{-1} goes to zero and the i -th thruster/actuator avoids saturation. Depending upon whether the thruster/actuator is approaching toward or departing from its saturation

limit, \hat{h}_i can be redefined as $\hat{h}_i = 1 + |\partial H(F_i)/\partial F_i|$ when $\Delta|\partial H(F_i)/\partial F_i| \geq 0$ (i.e., the thruster/actuator force/torque is approaching toward its limit), and $\hat{h}_i = 1$ when $\Delta|\partial H(F_i)/\partial F_i| \leq 0$ (i.e., the thruster/actuator force/torque is departing from its limit). Finally, the desired thruster/actuator force/torque vector, F_{td} , that is guaranteed to stay within the saturation limit.

Substituting Equation (31) into Equation (30) and denoting $e = q_d - q$, we obtain the following error equation in joint-space

$$\ddot{e} + K_1\dot{e} + K_2\dot{e} + K_3e = 0 \quad (36)$$

Thus, for positive values of the gains, K_1 , K_2 and K_3 , the joint-space errors reduce to zero asymptotically, as time goes to infinity.

Now, to incorporate both the fault and the saturation information in control Equation (31), we define a *fault-saturation matrix*, $\Gamma_{p \times p} = W_F^{-1}$, having diagonal elements either $1/\hat{h}_i$ or zero. Whenever there is any thruster/actuator fault, we put that corresponding diagonal element in Γ matrix as zero. If there is no fault in any thruster/actuator, that corresponding diagonal entry of the fault-saturation matrix will be $1/\hat{h}_i$. Thus, it accounts for the force/torque saturation limits along with the fault information. We can rewrite the Equation (31) in terms of fault-saturation matrix, Γ , as

$$v = \Gamma \eta^T (\eta \Gamma \eta^T)^{-1} [\ddot{q}_d + K_1(\ddot{q}_d - \ddot{q}) + K_2(\dot{q}_d - \dot{q}) + K_3(q_d - q)] - \gamma \quad (37)$$

5.2 Results and Discussion

We present the simulation results for a circular trajectory to demonstrate the effectiveness of the proposed method. In the simulation, an ODIN type vehicle (with higher thruster capacity) tries to track a circular path of diameter $2.65m$ in a horizontal plane in $20sec$. The vehicle attitudes are kept constant at $[0^0 \ 0^0 \ 90^0]$. We have considered three different cases for this circular trajectory tracking task. In Case 1, all thrusters are in working condition. In Case 2, two of the thrusters (Thruster 1 and 5) develop faults during operation. Case 3 is similar to Case 2 except, in this case, the thruster saturation limits are imposed. In Case 1, all thrusters are in working condition and therefore the thruster fault matrix, Ψ , becomes an identity matrix. In Case 2 and 3, Thruster 5 stops functioning after $7sec$ and Thruster 1 stops functioning after $12sec$. We have simulated it by incorporating zeros for the corresponding elements in the Ψ matrix.

It should be noted that the chosen circular trajectory tracking task is a much faster task (average speed = $0.808knot$, Fig. 15) compared to the straight-line trajectory tracking task (average speed = $0.032knot$) as discussed in Section 4.3. We wanted to see the performance of the proposed controller in a high-speed trajectory tracking with both thruster fault and thruster saturation. We could not risk the expensive ODIN for such a high-speed operation and thus, we provide only simulation results to demonstrate the efficacy of the proposed technique. Additionally, we could not experimentally verify the thruster saturation controller because ODIN was not equipped with any acceleration feedback mechanism.

We present the simulation results for the circular trajectory tracking task considering all the three cases: with no thruster fault, with thruster fault, and with thruster fault and thruster saturation in Fig. 15 and 16. We have simulated two thruster faults: one (Thruster 5, marked

by 'A' at 7sec and the other (Thruster 1, marked by 'B') at 12sec, Fig. 16. Both the faulty thrusters were chosen to be located at the same thruster bracket of the AUV. Thus, this fault was one of the worst fault conditions. We have imposed the following thruster saturation limits: $\pm 50N$ for vertical thrusters and $\pm 150N$ for horizontal thrusters. The task-space paths and trajectories are plotted in Fig. 15. It is observed that the trajectories are tracked quite accurately in all the three cases. However, the tracking errors are more for the thruster saturation case.

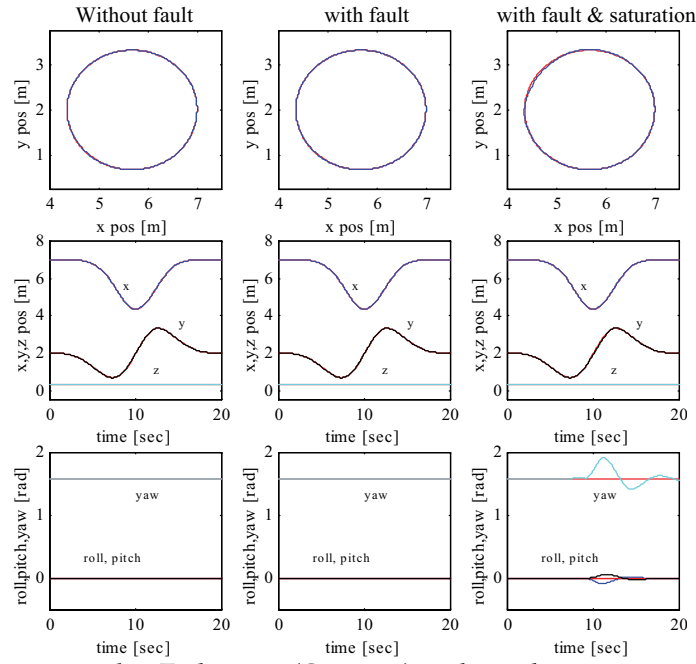


Fig. 15. Simulation results: Task-space (Cartesian) paths and trajectories, the solid lines denote the actual trajectories and the dashed lines denote desired trajectories.

The thruster forces are plotted in Fig. 16. From these plots, we can see that after the first fault, thrust for Thruster 7 becomes close to $-200N$. But by implementing the thruster saturation algorithm, we are able to keep this thrust within the specified limit ($\pm 150N$). In this process, the thrusts for Thruster 6 and Thruster 8 reach the saturation limits, but do not cross it. As a result, we observe larger errors in the trajectory tracking during this time for the saturation case (Fig. 15). However, the controller brings back the AUV in its desired trajectories and the errors are gradually reduced to zero.

6. Drag Minimization Algorithm

A UVMS is a kinematically redundant system. Therefore, a UVMS can admit an infinite number of joint-space solutions for a given task-space coordinates. We exploit this particular characteristic of a kinematically redundant system not only to coordinate the motion of a UVMS but also to satisfy a secondary objective criterion that we believe will be useful in underwater applications.

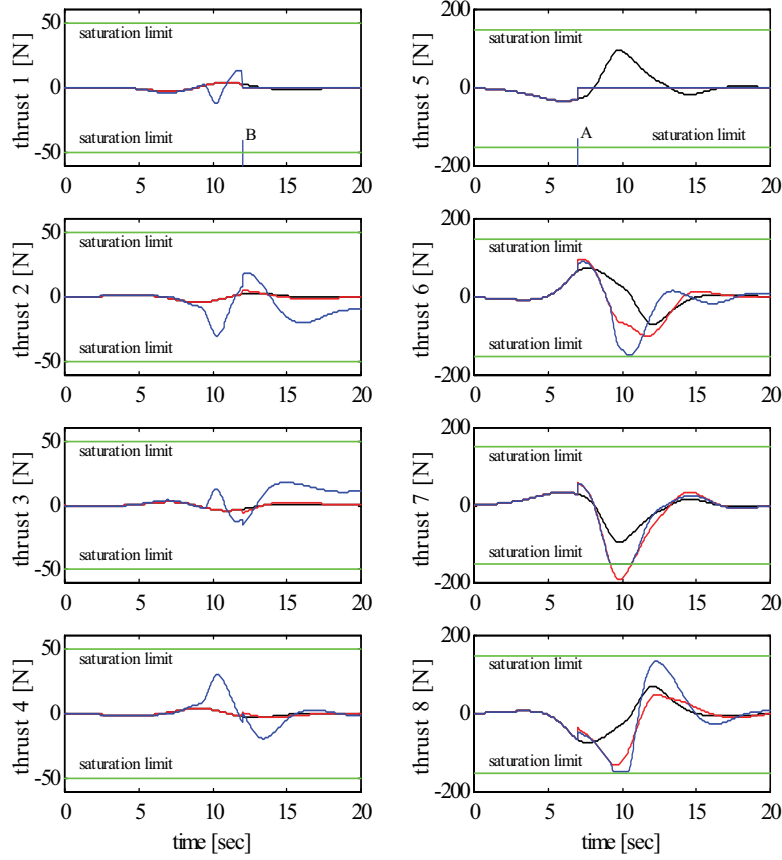


Fig. 16. Simulation results: Thrust versus time, no fault (Case 1, denoted by dashed-dot lines), with faults (Case 2, denoted by dashed lines), and saturation (Case 3, denoted by solid lines).

The secondary objective criterion that we choose to satisfy in this work is hydrodynamic drag optimization. Thus, we want to design a motion planning algorithm that generates trajectories in such a way that the UVMS not only reaches its goal position and orientation from an initial position and orientation, but also the drag on the UVMS is optimized while it follows the generated trajectories. Drag is a dissipative force that does not contribute to the motion. Actually, a UVMS will require a significant amount of energy to overcome the drag. Since the source of energy for an autonomous UVMS is limited, which generally comes from the batteries that the UVMS carries with it unlike a ROV and tele-manipulator system where the mother ship provides the energy, we focus our attention to reduce the drag on the system. Reduction of drag can also be useful from another perspective. The UVMS can experience a large reaction force because of the drag. High reaction force can saturate the controller and thus, degrade the performance of the system. This problem is less severe when human operators are involved because they can adjust their strength and coordination according to the situation. However, for an autonomous controller, it is better to reduce such a large force especially when the force is detrimental to the task.

6.1 Theoretical Development

Recalling Equation (11) and Equation (13), we can write the complete solution to the joint-space acceleration as (Ben-Israel & Greville, 1974):

$$\ddot{q}_{d_1} = J_{W_1}^+ (\ddot{x}_{d_1} - \dot{J} \dot{q}_{d_1}) + (I - J_{W_1}^+ J) \ddot{\phi}_1 \quad (38)$$

$$\ddot{q}_{d_2} = J_{W_2}^+ (\ddot{x}_{d_2} - \dot{J} \dot{q}_{d_2}) + (I - J_{W_2}^+ J) \ddot{\phi}_2 \quad (39)$$

where the null-space vectors $(I - J_{W_i}^+ J) \ddot{\phi}_i$ (for $i=1,2$) will be utilized to minimize the drag effects on the UVMS.

We define a positive definite scalar potential function $p(q, \dot{q})$, which is a quadratic function of drag forces as

$$p(q, \dot{q}) = D^T(q, \dot{q}) W_D D(q, \dot{q}) \quad (40)$$

where $D(q, \dot{q}) \in \mathbb{R}^{(6+n)}$ is the vector of drag forces and $W_D \in \mathbb{R}^{(6+n) \times (6+n)}$ is a positive definite weight matrix. Note that a proper choice of this W_D matrix can enable us to design the influence of drag on individual components of the UVMS. Generally, W_D is chosen to be a diagonal matrix so that the cross-coupling terms can be avoided. If it is chosen to be an identity, then the drag experienced on all dof of the combined system is equally weighted. However, increasing or decreasing the values of the diagonal elements of the W_D matrix, the corresponding drag contribution of each dof can be regulated. The potential function, $p(q, \dot{q})$, captures the total hydrodynamic drag on the whole vehicle-manipulator system. Therefore, the minimization of this function will lead to the reduction of drag on the whole system.

Now, taking the gradient of the potential function, $p(q, \dot{q})$, we obtain

$$\nabla p(q, \dot{q}) = \frac{\partial p(q, \dot{q})}{\partial q} + \frac{\partial p(q, \dot{q})}{\partial \dot{q}} \quad (41)$$

We take the gradient, $\nabla p(q, \dot{q})$, as the arbitrary vector, $\ddot{\phi}_i$, of Equation (38) and Equation (39) to minimize the hydrodynamic drag in the following form:

$$\ddot{\phi}_i = -\kappa_i \nabla p^T \quad \text{for } i = 1, 2. \quad (42)$$

where κ_i are arbitrary positive quantities, and the negative sign implies minimization of the performance criteria. A block diagram of the proposed control scheme is shown in Fig. 17. More detailed discussion on this drag minimization can be found in (Sarkar & Podder, 2001).

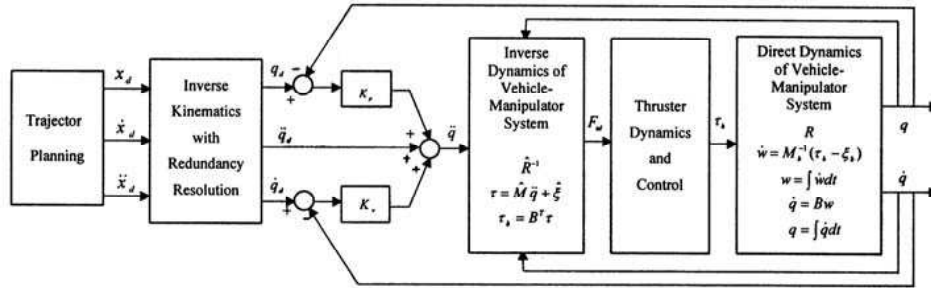


Fig. 17. Computer torque control scheme for drag minimization method.

6.2 Results and Discussion

We have conducted extensive computer simulations to investigate the performance of the proposed Drag Minimization (DM) algorithm. The details of the UVMS used for the simulation have been provided in Section 3.3. We have chosen a straight-line trajectory (length = $10m$) in the task-space for the simulations. For the chosen trajectory, we have designed a trapezoidal velocity profile, which imposes a constant acceleration in the starting phase, followed by a cruise velocity, and then a constant deceleration in the arrival phase. The initial velocities and accelerations are chosen to be zero and the initial desired and actual positions and orientations are same. The simulation time is $15.0sec$. Thus, the average speed of the UVMS is $0.67m/s \approx 1.30knot$. This speed is chosen to simulate the average speed of SAUVIM (Semi-Autonomous Underwater Vehicle for Intervention Mission), a UVMS being designed at the University of Hawaii, which has a maximum speed of $3knot$.

In our simulation, we have introduced sensory noise in position and orientation measurements. We have chosen Gaussian noise of $1mm$ mean and $1mm$ standard deviation for the surge, sway and heave position measurements, $0.1deg$ mean and $0.1deg$ standard deviation for the roll, pitch and yaw position measurements for the vehicle, and $0.01deg$ mean and $0.01deg$ standard deviation for the joint position measurements for the manipulator. We have also incorporated a 15% modeling inaccuracy during computer simulations to reflect the uncertainties that are present in underwater environment. This inaccuracy has been introduced to observe the effect of both the uncertainty in the model and the neglected off-diagonal terms of the added mass matrix.

Thruster dynamics have been incorporated into the simulations using the thruster dynamic model described later in Section 7.2. The thruster configuration matrix is obtained from the preliminary design of SAUVIM type UVMS. It has 4 horizontal thrusters and 4 vertical thrusters. The thruster configuration matrix for the simulated UVMS is as follows:

$$E = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -R_{t3} & 0 & R_{t3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & R_{t4} & 0 & -R_{t4} & 0 & 0 & 0 \\ R_{t1} & -R_{t2} & -R_{t1} & R_{t2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (43)$$

where $R_{t1} = 1.25m$, $R_{t2} = 1.75m$, $R_{t3} = 0.75m$, and $R_{t4} = 1.25m$ are the perpendicular distances from the center of the vehicle to the axes of the side and the front horizontal thrusters, and the side and the front vertical thrusters, respectively. The thrusters for UVMS are chosen to be DC brushless thrusters, model 2010 from TECNADYNE. The thruster propeller diameter is $0.204m$. It can produce approximately $580N$ thrust. The weight of each thruster is $7.9Kg$ (in water).

The simulation results are presented in Fig. 18 through Fig. 20. In Fig. 18, we have plotted both the desired and the actual 3D paths and trajectories. From the plots, it is observed that the end-effector of the manipulator tracks the desired task-space trajectories satisfactorily in both the PI and the DM methods. From the joint-space trajectories in Fig. 19, we can see that even though the UVMS follows the same task-space trajectories in both PI and DM methods, it does it with different joint-space configurations. This difference in joint-space configurations contributes to drag minimization as shown in Fig. 20. The total energy consumption of the UVMS has also been presented in Fig. 20. We find that the energy consumption is less in DM method as compared to that of in PI method. From these plots we observe that the drag on the individual components of UVMS may or may not be always smaller in DM method. But we can see in Fig. 20 that the total drag (norm of drag) on UVMS is less in DM method as compared to that of in PI method.

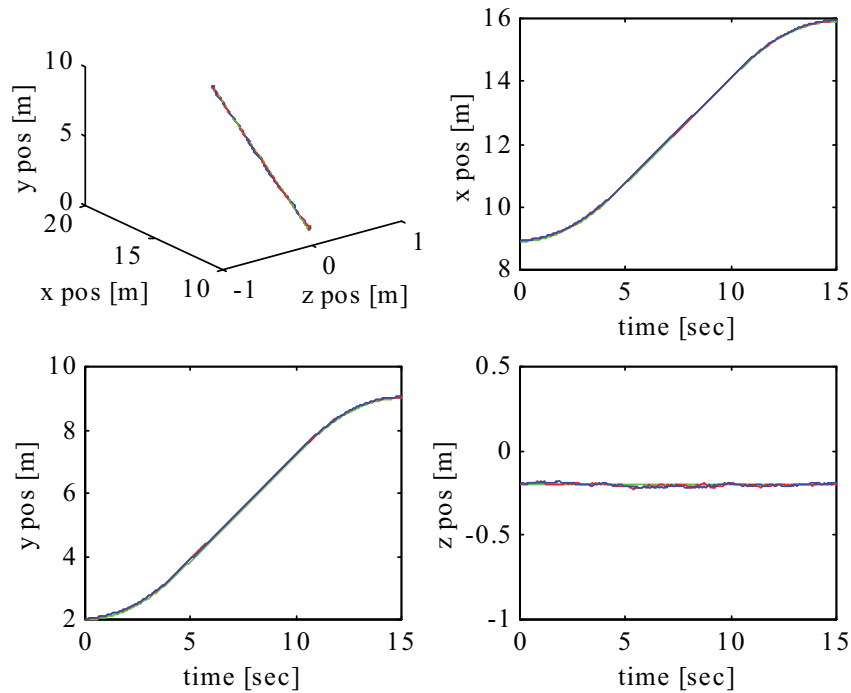


Fig. 18. Task-space (XYZ) straight-line trajectories of the end-effector of the robot manipulator, solid lines denote DM (drag minimization) method, dashed lines denote PI (pseudoinverse) method, and dashed dot lines denote the desired trajectories.

Here we have designed a model-based controller to follow a set of desired trajectories and presented results from computer simulations to demonstrate the efficacy of this newly proposed motion planning algorithm. In this context we must mention that a purely model-based controller may not be ideal for underwater applications. However, since the main thrust of this study is in motion planning, we have used this model-based controller only to compare the effectiveness of the proposed Drag Minimization algorithm with that of more traditionally used Pseudoinverse algorithm.

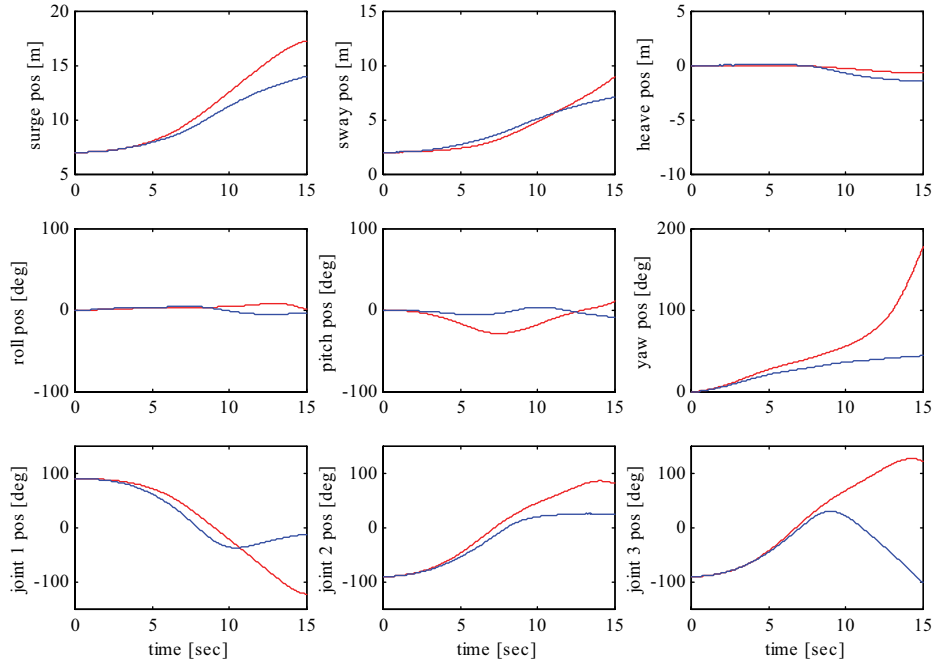


Fig. 19. Joint-space positions of the UVMS, solid lines denote DM (drag minimization) method, dashed lines denote PI (pseudoinverse) method.

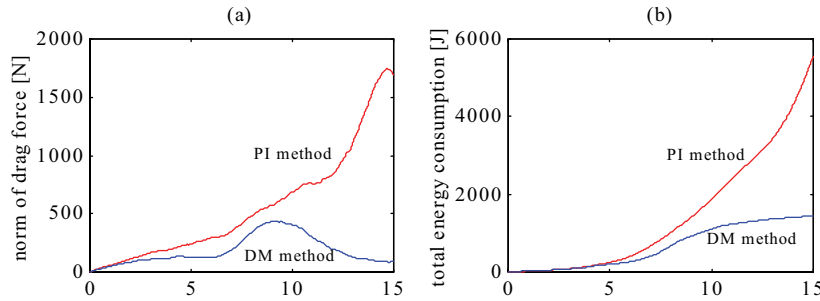


Fig. 20. (a) Norm of drag force, (b) total energy consumption of the UVMS, solid lines denote DM (drag minimization) method, dashed lines denote PI (pseudoinverse) method.

7. Unified Dynamics-Based Motion Planning Algorithm

7.1 Theoretical Development

A schematic diagram of the proposed unified dynamics-based control scheme is given in Fig. 21. For a unified dynamics-based algorithm, let us look back to Equations (38) and (39) along with Equations (40)-(42) which provide us with the reference joint-space trajectories considering the dynamics-based planning method as well as the drag minimization scheme. Now, we can obtain all the desired joint-space variables required for the control law (Eq. (31) or Eq. (37)) by integrating Equation (38) and Equation (39) and making use of Equation

(13). Then by differentiating it we can obtain the desired third derivative for the joint-space variables. Thus, we have formulated a unified motion planning algorithm by integrating the dynamics-based planning algorithm (Eq. (11)-(13)) with fault-tolerant algorithm (Eq. (29)), saturation algorithm (Eq. (30)-(31)), and drag minimization algorithm (Eq. (38)-(39)).

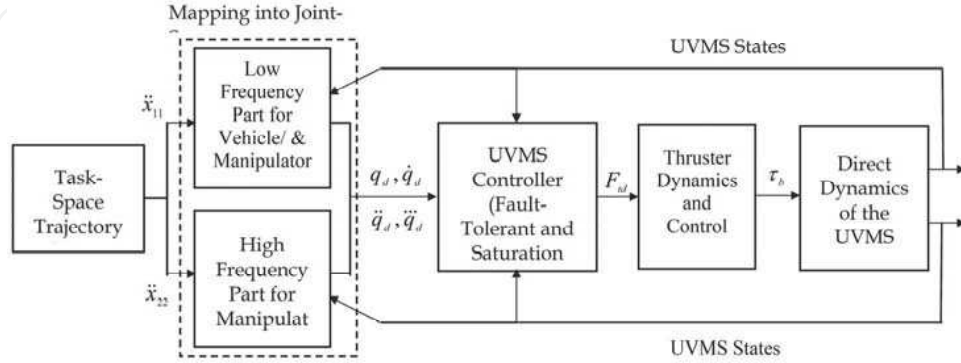


Fig. 21. Unified dynamics-based motion planning scheme.

7.2 Thruster Dynamics

The desired thruster force allocation as obtained from Equation (23) can be directly applied to the dynamic model of the UVMS given by Equation (1) (using Equation (15)) to generate the actual motion of the system. However, in such a case the dynamics of the thrusters will be neglected and the results will not accurately reflect the reality. Yoerger *et al.* (Yoerger *et al.*, 1990) pointed out that the system dynamics of an underwater vehicle can be greatly influenced by the dynamics of the thrusters, and neglecting this dynamics may result in a limited bandwidth controller with limit cycle instability. There are several dynamic models of marine thrusters (Yoerger *et al.*, 1990; Healey *et al.*, 1995; Whitcomb & Yoerger, 1999) that can reliably account for thruster dynamics.

In this work we use the model proposed by Healey *et al.* (Healey *et al.*, 1995) that included a four-quadrant mapping of the lifts and drag forces of the propeller blades and was coupled with the motor and fluid system dynamics. This model is given by the following equations:

$$\Omega_r = \alpha_2^{-0.5} \text{sign}(F_{td}) |F_{td}|^{0.5} \quad (44)$$

$$i_m = K_t^{-1} \alpha_1 F_{td} + K_f^{-1} K_{fb} (\Omega - \Omega_r) \quad (45)$$

$$\dot{\Omega} = I^{-1} [K_t i_m - K_f \Omega - \mathfrak{S}] \quad (46)$$

where Ω and Ω_r are the actual and the desired/reference propeller angular velocity, respectively, and i_m is the motor current. The other parameters are: $\alpha_2 = \rho A r^2 \eta^2 \tan^2(\gamma)$, where ρ is the density of the water, r is the radius of the propeller, A is the thruster duct area, η is the propeller efficiency, γ is the average pitch of the propeller blade, α_1 is an experimentally determined constant, K_t is the motor torque constant, K_f is the motor viscous friction constant, K_{fb} is the motor feedback gain, and \mathfrak{S} is the propeller shaft torque.

Neglecting the motor inductance (Healey *et al.*, 1995), the motor input voltage can be written as

$$V_m = i_m R_m + K_{emf} \Omega \quad (47)$$

where V_m is the motor input voltage, R_m is the motor resistance and K_{emf} is the motor back emf constant.

The propeller torque and the axial thrust are related to the blade lift, L and the drag, D as follows:

$$\mathfrak{T} = 0.7rL \sin \theta + D \cos \theta \quad (48)$$

$$F_{t,act} = L \cos \theta - D \sin \theta \quad (49)$$

where $F_{t,act}$ is the propeller shaft thrust, $\theta = \gamma - \alpha$, and α is the angle of attack.

7.3 Results and Discussion

We have performed extensive computer simulation to investigate the efficacy of the proposed Unified Dynamics-based Motion Planning (UDMP) algorithm. To verify the effectiveness of the proposed method, we have compared the results of UDMP approach with that of Conventional Motion Planning (CMP) method. In conventional method, the trajectory is designed in three sections: the main section (intermediate section) that is a straight line is preceded and followed by two short parabolic sections. The UVMS used for these simulations is same as mentioned in Section 3.3. The simulation time is 10sec that is required to complete the square path. The total length of the path is 8m, thus the average speed is about 1.6knot. This speed is close to JASON II vehicle (speed=1.5knot).

We have simulated two thruster faults: one horizontal thruster (Thruster 1) and the other one vertical thruster (Thruster 5). Both the thrusters stop functioning from 6sec. It is to be noted that both the thrusters are located at the same bracket of the UVMS, which is one of the worst thruster fault situations. In our simulation, we have considered the following thruster/actuator thrust/torque saturation limits: $\pm 400N$ for horizontal thrusters (Thruster 1-4), $\pm 200N$ for vertical thrusters (Thruster 5-8), $\pm 200N.m$ for actuator 1, $\pm 100N.m$ for actuator 2 and $\pm 50N.m$ for actuator 3.

To make the simulation close to reality, we have introduced sensory noise in the measurements of positions and its derivatives. We have considered Gaussian noise of 1 mean and 1 standard deviation in the measurement of linear quantities (in mm unit), and 0.01 mean and 0.05 standard deviation in measurement of angular quantities (in deg unit). We have considered 10% modeling inaccuracy during computer simulation to reflect the uncertainties that are present in underwater environment.

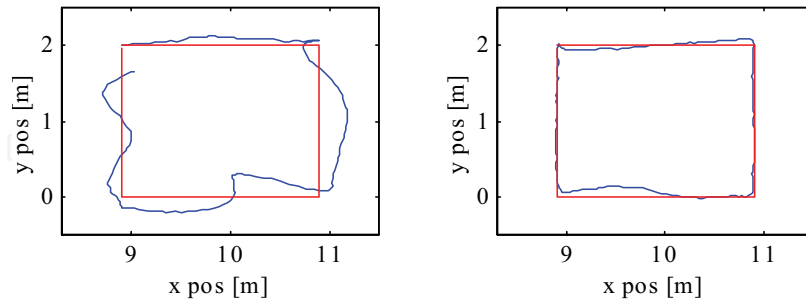


Fig. 22. Task-space geometric paths: Conventional Motion Planning (CMP) method in the left and Unified Dynamics-based Motion Planning (UDMP) method in the right. Doted lines denote the desired paths and solid lines denote actual paths.

We have presented results from the computer simulations in Fig. 22 through Fig. 26. The results we have provided here are from *Case I: Partial Decomposition* of the proposed UDMP method. The

task-space geometric paths are plotted in Fig. 22, where we can see that the path tracking errors in our proposed UDMP method are much smaller as compared to that of CMP method. We have also plotted task-space trajectories in Fig. 23. It is also observed from plots in Fig. 23 that the end-effector tracks the task-space trajectories quite accurately in UDMP method. The errors are less in proposed UDMP method as compared to the CMP method. The joint-space trajectories are plotted in Fig. 24. From these plots it is observed that the proposed UDMP method effectively reduces the motions of the heavy subsystem (the vehicle) and allows greater and sharper motions to the lighter subsystem (the manipulator) while tracking the same task-space trajectories. It is also noticed that the motion of the heavy subsystem is smoother in the proposed method. We find that these sharper and larger motions of the heavy subsystem in case of CMP method demand higher driving force that we see in Fig. 25. From the plots in this Fig. (Fig. 25) it is also observed that in case of UDMP method thrusters 4, 7, 8 and actuator 1 have reached the saturation limits, but they have not exceeded the limits. On the other hand, in case of CMP method all the thrusters and actuators have reached the saturation limits, however the saturation scheme was able to keep them to within the specified limits. Because of this, the path and trajectory tracking performance in CMP method has been degraded, as we can see in Fig. 22 and Fig. 23. Thus, the conventional planning method demands more powerful actuation system to track the same trajectories with reasonable accuracy. We also observe that the thrust 1 and thrust 5 are zero from 6th sec as marked by "A" and "B", respectively (see Fig. 25). These imply they have developed faults at 6th second and remain non-functional for rest of the time. At this moment we observe some perturbations in trajectories and paths, however, the proposed UDMP scheme gradually brings the system to the desired directions and reduces the tracking errors. On the other hand, after the occurrence of faults the paths and the trajectories are tracked poorly in case of CMP method, because this algorithm cannot account for the dynamics of the system while generating the reference trajectories.

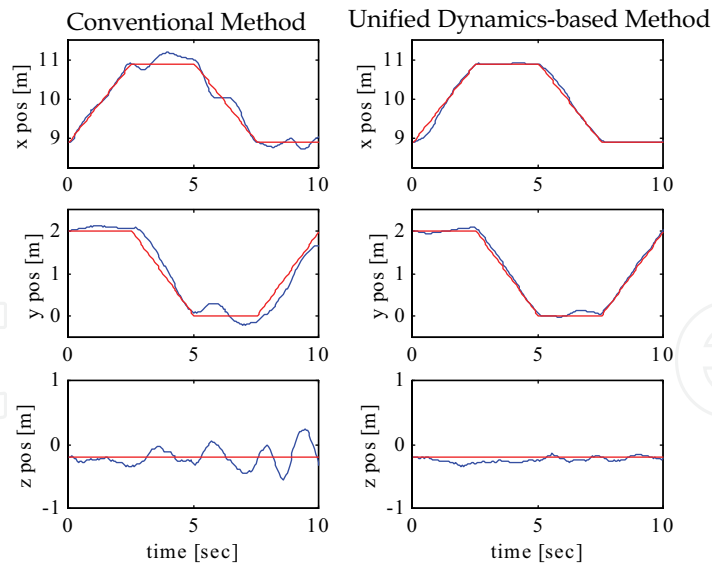


Fig. 23. Task-space trajectories: Conventional Motion Planning method (left column) and Unified Dynamics-based Motion Planning method (right column). Actual trajectories (solid lines) are superimposed on desired trajectories (dotted lines).

We have also plotted the simulation results for surge-sway motion, power requirement and energy consumption of the UVMS in case of CMP method (in the left column) and that of in case of proposed UDMP method (in the right column) in Fig. 26. Top two plots in this figure show the profile of the surge- sway movements of the vehicle in the said two methods. In case of the CMP method, the vehicle changes the motion sharply and moves more as compared to the motion generated from the UDMP method. It may so happen that, in practice, this type of sharp and fast movements may be beyond the capability of the heavy dynamic subsystem and consequently large errors in trajectory tracking will occur. Additionally, this may cause saturation of the thrusters and the actuators resulting in degradation in performance. Moreover, the vehicle will experience large velocity and acceleration in CMP method that result in higher power requirement and energy consumption, as we observe it in next two sets of plots in Fig. 26. Thus, this investigation reveals that our proposed Unified Dynamics-Based Motion Planning method is very promising for autonomous operation of dynamic system composed of several subsystems having variable dynamic responses.

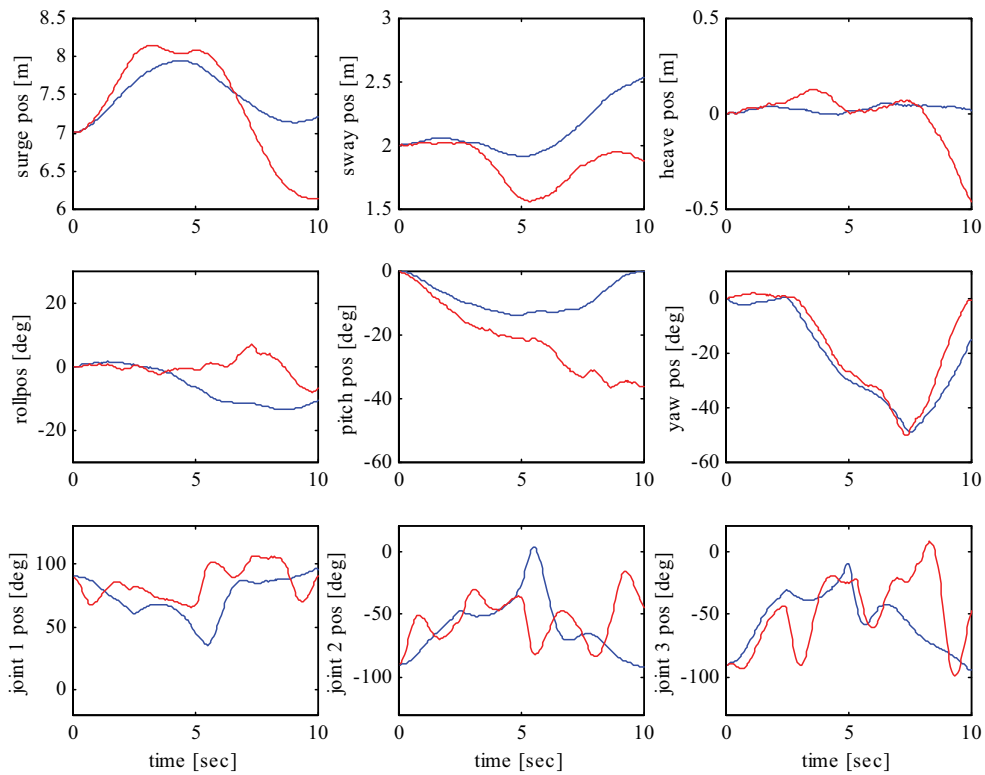


Fig. 24. Joint-space trajectories: Unified Dynamics-based Motion Planning method (solid lines) and Conventional Motion Planning method (dashed lines).

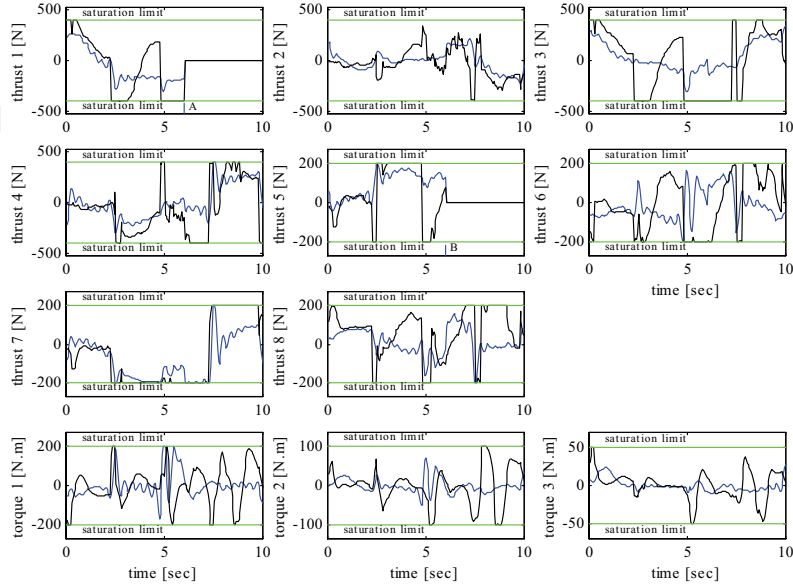


Fig. 25. Thruster and actuator forces and torques of the UVMS. Unified Dynamics-based Motion Planning method (solid lines) and Conventional Motion Planning method (dashed lines). Thruster faults are marked by “A” (Thruster 1) and “B” (Thruster 5).

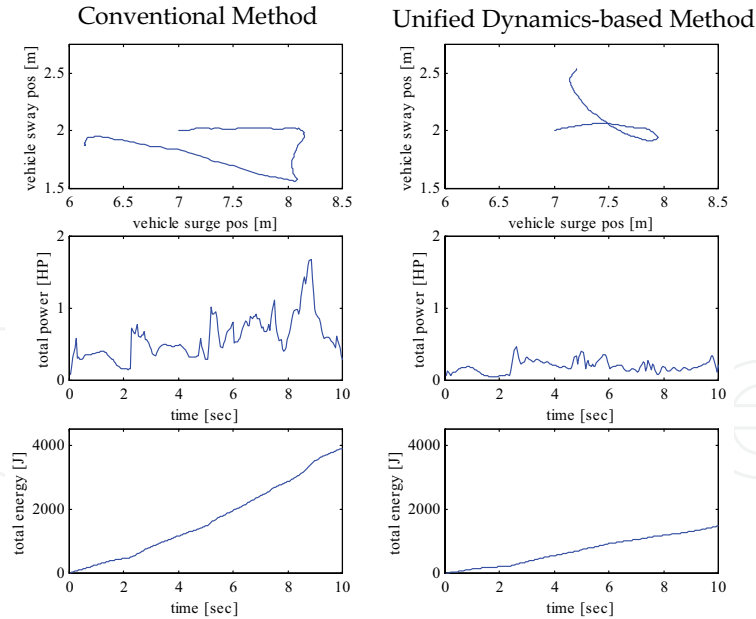


Fig. 26. Surge-sway motion of the vehicle, power requirement and energy consumption of the UVMS. Results from Conventional Motion Planning method are in the left and that of Unified Dynamics-based Motion Planning method are in the right.

8. Conclusions

We have proposed a new unified dynamics-based motion planning algorithm that can generate both kinematically admissible and dynamically feasible joint-space trajectories for systems composed of heterogeneous dynamics. We have then extended this algorithm for an autonomous underwater vehicle-manipulator system, where the dynamic response of the vehicle is much slower than that of the manipulator. We have also exploited the kinematic redundancy to accommodate the thruster/actuator faults and saturation and also to minimize hydrodynamic drag. We have incorporated thruster dynamics when modeling the UVMS. Although, some researchers have exploited kinematic redundancy for optimizing various criteria, but those work have mainly addressed to problems with land-based robotics or space-robotics. Hardly any motion planning algorithm has been developed for autonomous underwater vehicle-manipulator system. In this research, work we have formulated a new unified motion planning algorithm for a heterogeneous underwater robotic system that has a vastly different dynamic bandwidth. The results from computer simulation demonstrate the effectiveness of the proposed method. It shows that the proposed algorithm not only improves the trajectory tracking performance but also significantly reduce the energy consumption and the power requirements for the operation of an autonomous UVMS. We have not presented results from Case II (*Total Decomposition*) because of the length of the paper. However, these results are comparable to the conventional motion planning approach. In future, instead of Fourier decomposition, one can try to use wavelet approach to decompose the task-space trajectory into system's sub-component compatible segments.

There are a few drawbacks of this paper as well. We used a model-based control technique to evaluate our planning algorithm. However, the underwater environment is uncertain and we need to use adaptive control techniques in future. Although the fault-tolerant control algorithm has been experimentally verified, the other proposed algorithms need to be validated by experiments.

9. References

- Klein, C.A. & C.H. Huang, C.H. (1983). Review of pseudoinverse control for use with kinematically redundant manipulators, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-13, No. 3, pp. 245-250.
- Paul, R. (1979). Manipulator Cartesian path planning, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-9, No. 11, pp. 702-711.
- Zhou, Z.L. & Nguyen, C.C. (1997). Globally optimal trajectory planning for redundant manipulators using state space augmentation method, *Journal of Intelligent and Robotic Systems*, Vol. 19, No. 1, pp. 105-117.
- Siciliano, B. (1993). Closed-loop inverse kinematics algorithms for redundant spacecraft/manipulator systems, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 95-100, Atlanta, GA, May 1993.
- Antonelli, G. & Chiaverini, S. (1998). Task-priority redundancy resolution for underwater vehicle-manipulator systems, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 756-761, Leuven, Belgium, May 1998.

- Vukobratovic, M. & Kircanski, M. (1984). A dynamic approach to nominal trajectory synthesis for redundant manipulators, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC 14, No. 4, pp. 580-586, July 1984.
- Bobrow, J.E. (1988). Optimal robot path planning using the minimum-time criterion, *IEEE Journal of Robotics and Automation*, Vol. 4, No. 4, pp. 443-450, August 1988.
- Shiller, Z. & Dubowsky, S. (1989). Robot path planning with obstacles, actuator, gripper, and payload constraints, *International Journal of Robotics Research*, Vol. 8, No. 6, pp. 3-18, December 1989.
- Shin, K. & McKay, N. (1986). A dynamic programming approach to trajectory planning of robot manipulators, *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 6, pp. 491-500, June 1986.
- Hirakawa, A.R. & Kawamura, A. (1997). Trajectory planning of redundant manipulators for minimum energy consumption without matrix inversion, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2415-2420, Albuquerque, New Mexico, April 1997.
- Saramago, S.F.P. & Steffen Jr. V. (1998). Optimization of the trajectory planning of robot manipulators taking into account the dynamics of the system, *Journal of Mechanism and Machine Theory*, Vol. 33, No. 7, pp. 883-894, 1998.
- Zhu, Z.H.; Mayorga, R.V. & Wong, A.K.C. (1999). Dynamic robot manipulator trajectory planning for obstacle avoidance, *Journal of Mechanics Research Communications*, Vol. 26, No. 2, pp. 139-144, 1999.
- Faiz, N. & Agrawal, S.K. (2000). Trajectory planning of robots with dynamics and inequalities, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3977-3983, San Francisco, CA, April 2000.
- Macfarlane, S. & Croft, E.A. (2003). Jerk-bounded manipulator trajectory planning: design for real-time applications, *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 1, pp. 42-52, February 2003.
- Brock, O. & Khatib, O. (1999). High-speed navigation using the global dynamic window approach, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 341-346, Detroit, MI, May 1999.
- Huang, Q.; Tanie, K. & Sugano, S. (2000). Coordinated motion planning for a mobile manipulator considering stability and manipulation, *International Journal of Robotics Research*, Vol. 19, No. 8, pp. 732-742, August 2000.
- Yamamoto, Y. & Fukuda, S. (2002). Trajectory planning of multiple mobile manipulators with collision avoidance capability, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3565-3570, Washington D.C., May 2002.
- Yamashita, A.; Arai, T.; Jun, O. & Asama, H. (2003). Motion planning of multiple mobile robots for cooperative manipulation and transportation, *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 2, pp. 223-237, April 2003.
- Yoerger, D.R. & Slotine, J.J.E. (1985). Robust trajectory control of underwater vehicles, *IEEE Journal of Oceanic Engineering*, Vol. 10, No. 4, pp. 462-470, October 1985.

- Spangelo, I. & Egeland, O. (1994). Trajectory planning and collision avoidance for underwater vehicle using optimal control, *IEEE Journal of Oceanic Engineering*, Vol. 19, No. 4, pp. 502-511, 1994.
- Kawano, K. & Ura, T. (2002). Motion planning algorithm for nonholonomic autonomous underwater vehicle in disturbance using reinforced learning and teaching method," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4032-4038, Washington D.C., May 2002.
- Sarkar, N. & Podder, T.K. (2001). Coordinated motion planning and control of underwater vehicle-manipulator systems subject to drag optimization," *IEEE Journal of Oceanic Engineering*, Vol. 26, No. 2, pp. 228-239, 2001.
- Fu, K.S. Gonzalez, R.C. & Lee, C.S.G. (1988). *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill International Edition, 1988.
- Craig, J.J. (1989). *Introduction to Robotics: Mechanics and Control*, Addison-Wesley Publishing Company, 1989.
- Kane, T.R. & Lavinson, D.A. (1985). *Dynamics: Theory and Applications*, McGraw-Hill Book Co., 1985.
- Fossen, T.I. (1984). *Guidance and Control of Ocean Vehicles*, John Willey & Sons Ltd., 1994.
- Podder, T.K. (2000). *Motion Planning of Autonomous Underwater Vehicle-Manipulator Systems*, Ph.D. Thesis, Department of Mechanical Engineering, University of Hawaii at Manoa, Honolulu, December 2000.
- James, M.L.; Smith, G.M.; Wolf, J.C. & Whaley, P.W. (1989). *Vibration of Mechanical and Structural Systems*, Harper and Row Publishers, New York, 1989.
- Podder, T.K. & Sarkar, N. (2000). Fault tolerant control of an autonomous underwater vehicle under thruster redundancy: simulation and experiments, *Proceedings of IEEE International Conference on Robotics and Automation*, pp.1251 -1256, San Francisco, April 2000.
- Podder, T.K.; Antonelli, G. & Sarkar, N. (2001). An experimental investigation into the fault-tolerant control of an autonomous underwater vehicle, *Journal of Advanced Robotics*, Vol. 15, No. 5, pp. 501-520, 2001.
- Isidori, A.; Moog, C.H. & De Luca, A. (1986). A sufficient condition for full linearization via dynamic state feedback, *Proceedings of IEEE Conference on Decision and Control*, pp. 203-208, Athens, Greece, December 1986.
- Yun, X. (1988). Dynamic state feedback control of constrained robot manipulators, *Proceedings of IEEE Conference on Decision and Control*, pp. 622-626, Austin, Texas, December 1988.
- Ben-Israel, A. & Greville, T.N.E. (1974). *Generalized Inverse: Theory and Applications*, New York, Wiley, 1974.
- Yoerger, D.R.; Cooke, J. & Slotine, J.J.E. (1990). The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design, *IEEE Journal of Oceanic Engineering*, Vol. 15, No. 3, pp. 167-178, 1990.
- Healey, A.J.; Rock, S.M.; Cody, S. Miles, D. & Brown, J.P. (1995). Toward an improved understanding of thruster dynamics for underwater vehicles, *IEEE Journal of Oceanic Engineering*, Vol. 20, No. 4, pp. 354-361, 1995.

Whitcomb, L.L. & Yoerger, D.R. (1999). Development, comparison, and preliminary validation of non-linear dynamic thruster models, *IEEE Journal of Oceanic Engineering*, Vol. 24, No. 4, pp. 481-493, 1999.



Mobile Robots: Perception & Navigation

Edited by Sascha Kolski

ISBN 3-86611-283-1

Hard cover, 704 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, February, 2007

Published in print edition February, 2007

Today robots navigate autonomously in office environments as well as outdoors. They show their ability to beside mechanical and electronic barriers in building mobile platforms, perceiving the environment and deciding on how to act in a given situation are crucial problems. In this book we focused on these two areas of mobile robotics, Perception and Navigation. This book gives a wide overview over different navigation techniques describing both navigation techniques dealing with local and control aspects of navigation as well as those handling global navigation aspects of a single robot and even for a group of robots.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tarun K. Podder and Nilanjan Sarkar (2007). Unified Dynamics-based Motion Planning Algorithm for Autonomous Underwater Vehicle-Manipulator Systems (UVMS), Mobile Robots: Perception & Navigation, Sascha Kolski (Ed.), ISBN: 3-86611-283-1, InTech, Available from:
http://www.intechopen.com/books/mobile_robots_perception_navigation/unified_dynamics-based_motion_planning_algorithm_for_autonomous_underwater_vehicle-manipulator_syste

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2007 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen