15

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

Source: Mobile Robots: Perception & Navigation, Book edited by: Sascha Kolski, ISBN 3-86611-283-1, pp. 704, February 2007, Plv/ARS, Germany

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:

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

where the subscript *'b'* denotes the corresponding parameters in the body-fixed frames of the vehicle and the manipulator. $M_b(q_m) \in \Re^{(6+n)\times(6+n)}$ is the inertia matrix including the added mass and $C_b(q_m, w) \in \Re^{(6+n)\times(6+n)}$ is the centrifugal and Coriolis matrix including terms due to added mass. $D_b(q_m, w) \in \Re^{(6+n)\times(6+n)}$ is the drag matrix, $G(q) \in \Re^{(6+n)}$ is the vector of restoring forces and $\tau_b \in \Re^{(6+n)}$ is the vector of forces and moments acting on the UVMS. The displacement vector $q = [q_{v,} q_m]^T$, where $q_v = [q_{1,}, ..., q_6]^T$, and $q_m = [q_7, ..., q_{6+n}]^T$. 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. $q_7, q_8, ..., q_{6+n}$ are the angular displacements of joint 1, joint 2, ……., joint *n* of the manipulator in link-fixed frames. The quasi velocity vector $w = [w_1, ..., w_{6+n}]^T$, 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. $w_7, w_8, ..., w_{6+n}$ 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).

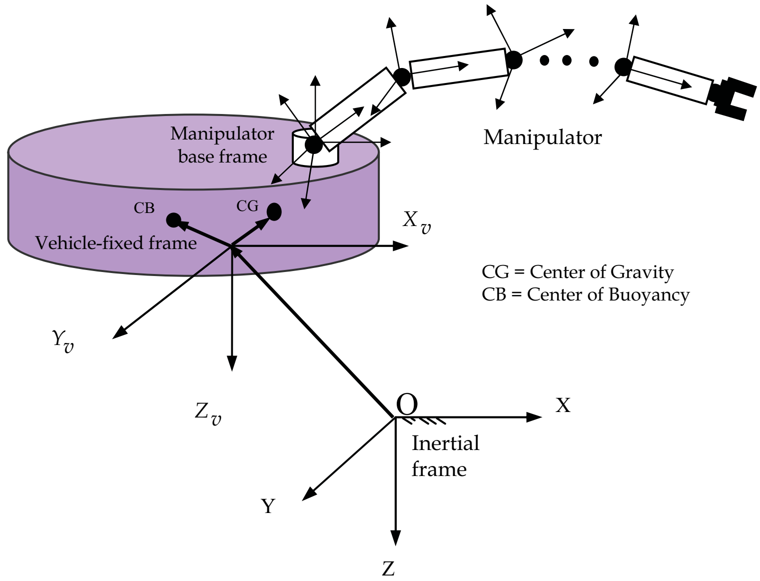

Fig. 1. Coordinate frames for underwater vehicle-manipulator system.

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:

$$\dot{q} = Bw \tag{2}$$

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

$$B(q) = \begin{bmatrix} B_{1_{6\times 6}} & O_{6\times n} \\ O_{n\times 6} & B_{2_{n\times n}} \end{bmatrix}_{(6+n)\times(6+n)}, \; B_1 = \begin{bmatrix} J_1 & O \\ O & J_2 \end{bmatrix}, \; B_2 = [I] \tag{3}$$

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

$$J_1 = \begin{bmatrix} C_5C_6 & -S_6C_4 + S_4S_5C_6 & S_4S_6 + S_5C_4C_6 \\ S_6C_5 & C_4C_6 + S_4S_5S_6 & -S_4C_6 + S_5S_6C_4 \\ -S_5 & S_4C_5 & C_4C_5 \end{bmatrix} \tag{4}$$

$$J_2 = \begin{bmatrix} 1 & S_4T_5 & C_4T_5 \\ 0 & C_4 & -S_4 \\ 0 & S_4/C_5 & C_4/C_5 \end{bmatrix} \tag{5}$$

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 $\pm 90^0$. Generally, the pitch angle in practical operation is restricted to $|q_5| < 90^0$. However, if we need to avoid singularity altogether, unit quarternions 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 ω_i, $i = 1,2,\cdots\cdots,s$.
We approximate the task-space trajectories using Fourier series and represent it in terms of the summation of several frequencies in ascending order.

$$x_{d\,6\times1}(t) = f_{6\times1}(t) = a_0 + \sum_{r=1}^{\infty} a_r \cos(r\pi t / L) + \sum_{r=1}^{\infty} b_r \sin(r\pi t / L) \tag{6}$$

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:

$$x_{d\ 6\times1}(t) = f_{6\times1}(t) = f_1(t) + f_2(t) + \cdots\cdots + f_{p_1}(t) \tag{7}$$

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 $j = 2,3,..., p_1$.
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:

$$\dot{q}_{d_j} = J^{+}_{W_j} \dot{x}_{d_j} \tag{8}$$

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

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 $j = 1,2,...,p_1$. $J^+_{w_j} = W_j^{-1} J^T (J W_j^{-1} J^T)^{-1}$ are the weighted pseudoinverse of Jacobians and $W_j = diag(h_{1_j}, ..., h_{(6+n)_j})$ 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 $(\omega_t \le \omega_i)$ 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:

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

where $f_{11}(t) = a_0 + \sum_{r=1}^{r_1} a_r \cos(r\pi t/L) + \sum_{r=1}^{r_1} b_r \sin(r\pi t/L)$, $f_{22}(t) = \sum_{r=r_1+1}^{r_2} a_r \cos(r\pi t/L) + \sum_{r=r_1+1}^{r_2} b_n \sin(r\pi t/L)$, 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

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

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

$$\ddot{q}_d = \ddot{q}_{d_1} + \ddot{q}_{d_2} \tag{13}$$

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

$$W_1 = diag(h_1, h_2, ..., h_{6+n}) \tag{14}$$

$$W_2 = diag(0, ..., 0, h_7, ..., h_{6+n}) \tag{15}$$

For Case II- Total decomposition:

$$W_1 = diag(h_1, ..., h_6, 0, ..., 0) \tag{16}$$

$$W_2 = diag(0, ..., 0, h_7, ..., h_{6+n}) \tag{17}$$

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,…...,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.

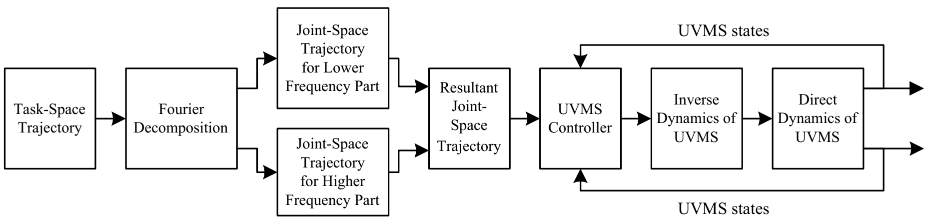

Fig. 2. Dynamics-based planning scheme.

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 ($\widetilde{P}$), (7) Using Rayleigh damping equation find a proportional damping matrix, and (8) Decouple the dynamic equations by using $\widetilde{P}$.
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 2.0*m*, 1.0*m* and 1.0*m*, respectively. The mass of the vehicle is 1073.0*Kg*. The links are cylindrical and each link is 1.0*m* long. The radii of link 1, 2 and 3 are 0.1*m*, 0.08*m* and 0.07*m*, respectively. The link masses (oil filled) are 32.0*Kg*, 21.0*Kg* and 16.0*Kg*, 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.

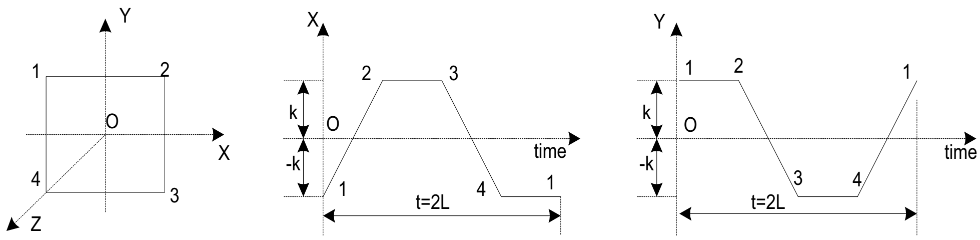

Fig. 3. Task-space geometric path and trajectories.

The task-space trajectories can be represented as

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

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

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

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:

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

For this simulation, we have taken *k= 1m,* i.e., the path is 2*m* 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 10.0*sec,* which is required to complete the square path in XY (horizontal) plane. The total length of the path is 8.0*m*; the average speed is about 1.6*knot*. This speed is more than JASON vehicle (speed = 1.0*knot*) but less than SAUVIM system (designed speed = 3.0*knot*).

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.

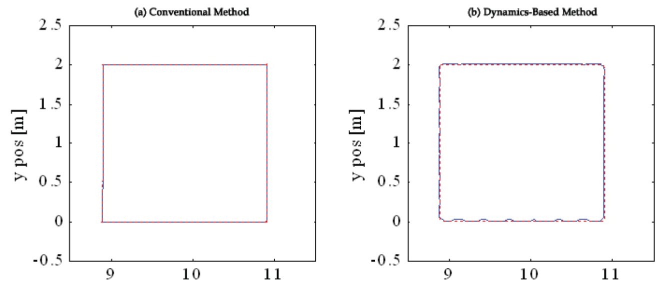

Fig. 4. Task-space geometric paths, (a) Conventional Straight-line planning method and (b) Dynamics-Based planning method for Case I. The actual path is denoted by solid line and the desired path is denoted by dashed line.

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.

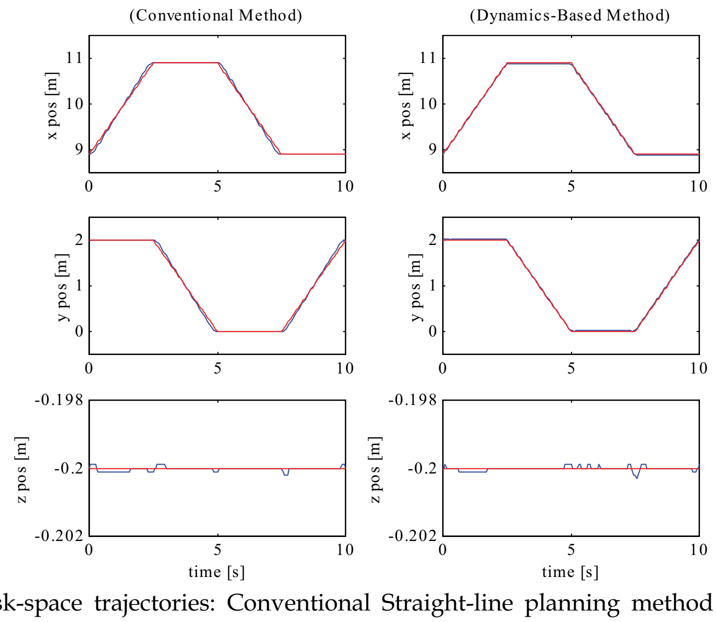

Fig. 5. Task-space trajectories: Conventional Straight-line planning method (left column) and Dynamics-Based planning method for Case I (right column). Desired trajectories are denoted by dashed lines and actual trajectories are denoted by dashed lines.

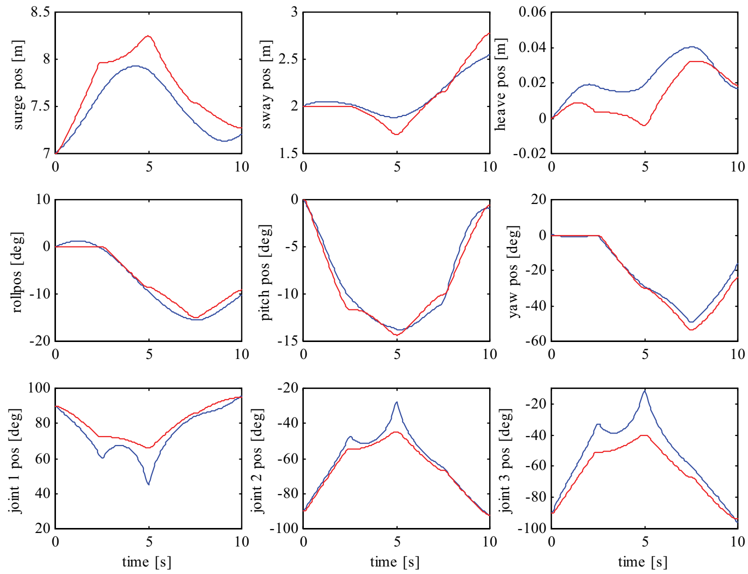

Fig. 6. Joint-space trajectories: Dynamics-Based planning method for Case I (solid/blue line) and Conventional Straight- line planning method (dashed/red line).

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.

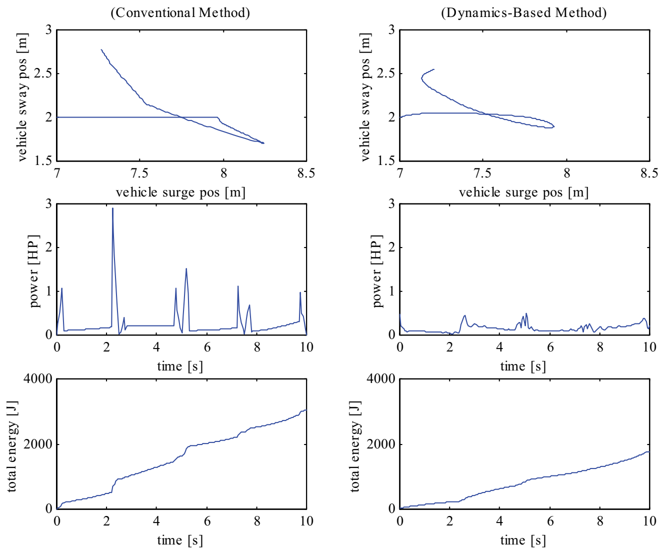

Fig. 7. X-Y motion of the center of gravity, power and energy consumption of the UVMS. Left column for Conventional Straight-line planning method and right column for Dynamics-Based planning method for Case I (partial decomposition).

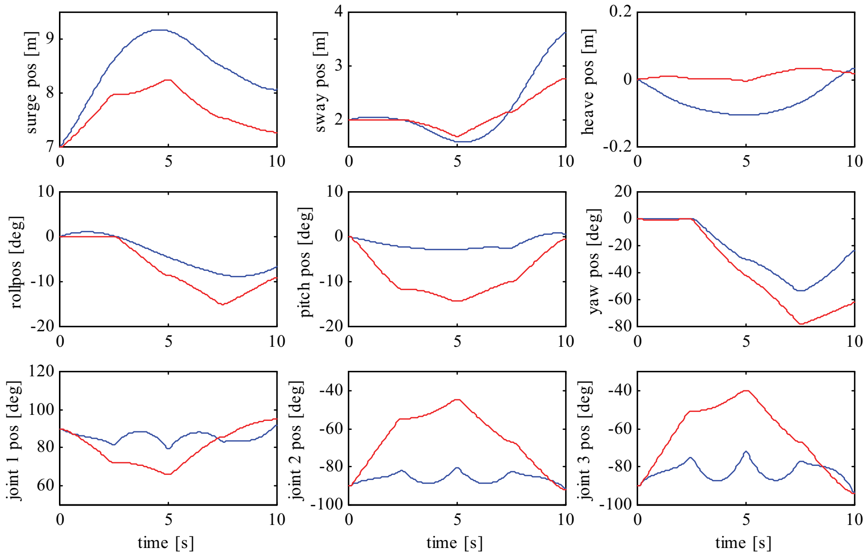

Fig. 8. Joint-space trajectories: Dynamics-Based planning method (solid/blue line) for Case II and Conventional Straight- line planning method (dashed/red line).

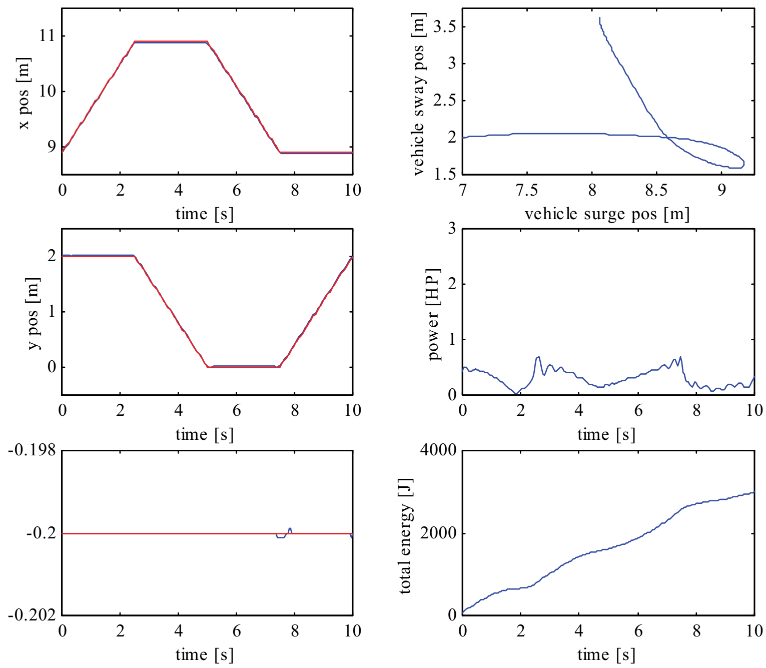

Fig. 9. Task-space trajectories (left column), and surge-sway motion, power requirement and energy consumption (right column) for Dynamics-Based planning method for Case II (total decomposition).

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

$$\tau_b = EF_t \tag{24}$$

where $E \in \Re^{(6+n)\times p}$ *thruster configuration matrix* and $F_t \in \Re^p$ 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

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

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:

$$\ddot{q} = B\dot{w} + \dot{B}w \tag{26}$$

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

$$\ddot{q} = \eta F_t + \lambda \tag{27}$$

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

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

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

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

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

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).

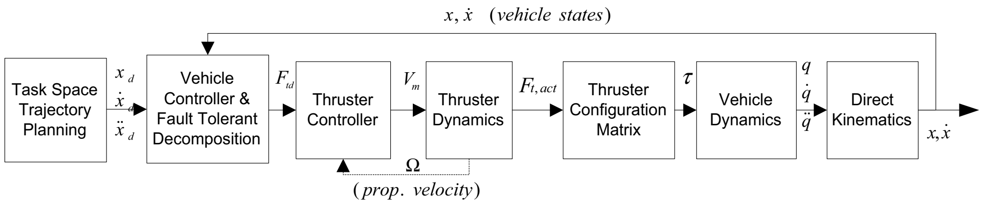

Fig. 10. Fault-tolerant control scheme.

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.

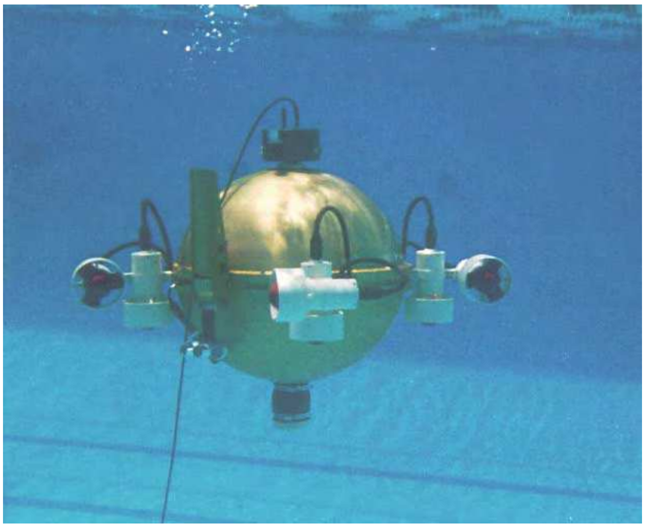

Fig. 11. Omni-Directional Intelligent Navigator (ODIN) vehicle.

The ODIN has a near-spherical shape with horizontal diameter of 0.63*m* and vertical diameter of 0.61*m*, made of anodized Aluminum (AL 6061-T6). Its dry weight is 125.0*Kg* and is slightly positively buoyant. The processor is a Motorola 68040/33*MHz* working with VxWorks 5.2 operating systems. The RS232 protocol is used for RF communication. The RF Modem has operating range up to 480*m*, operating frequency range 802-928*MHz*, 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 1*Kg* and can provide a maximum thrust of approximately 27*N*. The sensor system is composed of: 1) a pressure

sensor for measuring depth with an accuracy of 3*cm*, 2) 8 sonars for position reconstruction and navigation, each with a range of 0.1-14.4*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 100*sec* 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 120*sec* to reach a depth of 2*m*. Then it moves toward the y-direction for another 120.0*sec* to traverse 2.0*m*. It subsequently moves towards the x-direction for 120*sec* to traverse 2*m*. Finally it hovers at that position for another 40*sec*. ODIN follows a trapezoidal velocity profile during this task. The attitudes are always kept constant at $[0^0\ 0^0\ 90^0]$. For Case 2, one horizontal thruster (Thruster 6) fails at 260*sec* and one vertical thruster (Thruster 2) fails at 300*sec* 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 2*mm* mean and 1.5*mm* standard deviation for the surge, sway and heave position measurements, 0.15*degree* mean and 0.15*degree* 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 100.0*sec* before it works properly. Thus, x and y trajectories in experiments have data after 100.0*sec*. 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 260.0*sec* (marked as 'A') and one vertical thruster (Thruster 2) fails at 300.0*sec* (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.