Robotics and Autonomous Systems 34 (2001) 39–52

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Fault-tolerant control of an autonomous underwater vehicle under thruster redundancy

Tarun Kanti Podder [a], Nilanjan Sarkar [b,*]

[a] Department of Mechanical Engineering, University of Hawaii at Manoa, Honolulu, HI 96822, USA

[b] Department of Mechanical Engineering, Vanderbilt University, Box 1592, Station B, Nashville, TN 37235, USA

Received 20 December 1998; received in revised form 27 March 2000

Communicated by T.C. Henderson

Abstract

A novel approach to the allocation of thruster forces of an autonomous underwater vehicle (AUV) is investigated. Generally, the number of thrusters in an AUV is more than what is minimally required to produce the desired motion. We investigate how to exploit the excess number of thrusters to accommodate thruster faults during operation. First, a redundancy resolution scheme is presented that takes into account the presence of excess number of thrusters along with any thruster faults and determines the reference thruster forces to produce the desired motion. These reference thruster forces are utilized in the controller to generate the required motion. This approach resolves the thruster redundancy in the Cartesian space and allows the AUV to track the task-space trajectories with asymptotic reduction of the task-space errors. Results from computer simulations are presented to demonstrate the viability of the proposed scheme.

Keywords: Autonomous underwater vehicle; Fault-tolerant control; Thruster redundancy; Thruster force allocation; Underwater robotic system

1. Introduction

In recent years, the need and desire to explore the vast oceanic environment for extraction of resources and preservation of global environment have provided a significant momentum to underwater robotics research. Underwater robotic vehicle such as JASON at Woods Hole Oceanographic Institution and Ventana at the Monterey Bay Aquarium Research Institute have demonstrated the potential to enhance our ability to explore the ocean. However, remotely operated vehicles (ROVs) such as these are expensive because they require a large support vessel and experienced crew. Moreover, if tele-manipulation, which is the current practice for underwater task execution, is to be replaced by autonomous manipulation in future, the first step is to develop autonomous underwater vehicle (AUV) technology. The dynamics and control of an AUV have been investigated in recent years. Yoerger and Slotine [1] proposed a sliding-mode controller for robust trajectory following tasks. Goheen and Jeffery [2] investigated a nonlinear controller as an autopilot for an underwater vehicle for auto-positioning and station-keeping. Nakamura

* Corresponding author.

E-mail addresses: tarun@wiliki.eng.hawaii.edu (T.K. Podder), nilanjan.sarkar@vanderbilt.edu (N. Sarkar).

0921-8890/01/$ – see front matter
PII: S0921-8890(00)00100-7

and Savant [3,4] analyzed the control of a nonholonomic underwater vehicle based on a kinematic model. Healey and Marco [5] described a slow speed flight control of AUVs using traditional and sliding-mode controllers, and presented experimental results. Healey and Lienard [6] presented a multivariable sliding-mode controller for unmanned underwater vehicles. Fossen [7] described the use of multivariable sliding-mode control in dynamic positioning of ROVs. Adaptive controller [8] and learning controller based on neural network [9,10] were also proposed in the literature to accommodate the uncertainties in the underwater environment. A nonlinear optimal control approach based on Hamilton–Jacobi–Bellman equation for an AUV was presented recently [11]. Work on the modeling and control of thruster dynamics is also reported in the literature [12–14].

An AUV is expected to work in an unstructured environment. Therefore, it is necessary to develop a fault-tolerant control system for an AUV. A fault-tolerant system comprises of three modules: fault detection, fault isolation and fault accommodation. There are two major methods for fault detection. One of them is a model-free method, which relies on measured signals [15,16]. The other method is called model-based method where model-based analytical techniques are used to detect faults [17–19]. Fault isolation is concerned with the identification of the cause and location of the fault [19,20]. Fault accommodation allows the controller to execute its tasks in the presence of faults. There are several approaches to fault accommodation including analytical methods [21] and methods based on artificial intelligence [22–25]. The work on the thruster force allocation, on the other hand, is rare [26]. Generally, there are more thrusters than what is minimally required for the specific degrees-of-freedom (dof) that an AUV is designed for. This is done mainly to provide additional power and to enhance the ability of the AUV to perform its tasks in case of thruster faults. However, the introduction of additional thrusters complicates the allocation of thruster forces because of redundancy.

In this paper, we formulate the problem in the Cartesian space and provide a mathematical framework that enables the desired (reference) thruster force allocation based on a minimum norm criterion. We then proceed to further develop the framework to incorporate thruster faults. Finally, a Cartesian space control law that allows the AUV to follow the desired task-space trajectories is introduced. We verify the controller by extensive computer simulations.

This paper is organized as follows. In Section 2 we present the theoretical framework which forms the basis for the thruster force allocation. In Section 3 we provide results from computer simulations to demonstrate the viability of this proposed approach. Finally we summarize our contributions in Section 4.

2. Theoretical framework

In this section we present a mathematical formulation for the allocation of thruster forces of an AUV.

2.1. Dynamics

The dynamic equation of motion of an AUV can be written in the following form [27]:

$$M\dot{w} + C(w)w + D(w)w + G(q) = \tau, \tag{1}$$

where $q_{6\times1}$ are the generalized coordinates denoting the position and orientation of the AUV with respect to an inertial frame and $w_{6\times1}$ is the vector of linear and angular velocities of the AUV with respect to the vehicle's body-attached frame. $M_{6\times6}$ is the inertia matrix of the AUV which includes both the rigid body and the added mass terms. $C_{6\times6}$ is the matrix of centrifugal and Coriolis terms which includes both the rigid body and the added mass terms. $D_{6\times6}$ is the matrix of hydrodynamic drag terms and $G_{6\times1}$ is the vector of restoring forces (gravity and buoyancy). Finally, $\tau_{6\times1}$ is the vector of generalized forces on the AUV that is supplied by the thrusters.

The derivatives of the generalized coordinates, $\dot{q}$, and the linear and angular velocities of the vehicle in the body-attached frame, w, are related by the following equations [27]:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} = \begin{bmatrix} J_1(q_4, q_5, q_6)_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & J_2(q_4, q_5, q_6)_{3\times 3} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{bmatrix}. \tag{2}$$

In the above equation, $\dot{q}_1$, $\dot{q}_2$, and $\dot{q}_3$ represent the linear velocities of the AUV in the x, y, and z directions, respectively, of the inertial frame. $\dot{q}_4$, $\dot{q}_5$, and $\dot{q}_6$ represent the derivative of the Euler angles. w_1, w_2, and w_3 are the linear velocities (surge, sway, heave) of the AUV in the body-attached x, y, and z directions, respectively. w_4, w_5, and w_6 are the angular velocities (roll, pitch, yaw) of the AUV in the body-attached x, y, and z directions, respectively.

For roll–pitch–yaw Euler angles, the expressions for J_1 and J_2 are as follows:

$$J_1 = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi)c(\phi) + c(\psi)s(\theta)s(\phi) & s(\psi)s(\phi) + c(\psi)c(\phi)s(\theta) \\ s(\psi)c(\theta) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & -c(\psi)s(\phi) + s(\psi)c(\phi)s(\theta) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix},$$

$$J_2 = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \dfrac{s(\phi)}{c(\theta)} & \dfrac{c(\phi)}{c(\theta)} \end{bmatrix}.$$

Here ϕ, θ, and ψ are the roll, pitch and yaw angles of the AUV, respectively. The symbol $c(\cdot)$, $s(\cdot)$, and $t(\cdot)$ imply $\cos(\cdot)$, $\sin(\cdot)$, and $\tan(\cdot)$, respectively. Note that there is an Euler angle singularity in J_2 when pitch angle, θ, is an odd multiple of 90°. Generally, in practical applications, the pitch angle is kept at $|\theta| < 90°$. However, if we need to avoid the singularity altogether, unit quarternions can be used.

2.2. Thruster force allocation

In order to relate the generalized force vector τ with the individual thruster forces, let us consider an AUV which has n thrusters, where $n > 6$. In such a case, we can write

$$\tau = EF_{\text{td}}, \tag{3}$$

where $E_{6\times n}$ is the *thruster configuration matrix* and $F_{\text{td}_{n\times 1}}$ is the desired (reference) of thruster forces. The matrix E captures the geometry of the AUV and its thruster locations to transform the individual thruster force into generalized forces in the body-attached frame of the AUV. It is to be noted here that the matrix E is a constant matrix whose elements are the coordinates of the location of each thruster with respect to the center of mass of the AUV. Thus E is a constant matrix for a fixed geometry of the thrusters. The mechanical design of an AUV must ensure that E has rank 6 so that the AUV can generate forces and torques for three translational and three rotational motions. If its rank falls below 6, then the AUV will lose one or more of its dof. In this work, we assume that the AUV is designed in such a way that the thruster forces can span a six-dimensional space to produce the necessary motion in all 6 dof.

Substituting Eq. (3) into Eq. (1) and performing algebraic manipulation, we get

$$\dot{w} = M^{-1}(EF_{\text{td}} - \zeta), \tag{4}$$

where $\zeta = C(w) + D(w) + G(q)$.

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

$$\dot{q} = Bw, \tag{5}$$

where $B_{6\times 6}$ is a transformation matrix.

Differentiation of Eq. (5) leads to the following acceleration relationship:

$$\ddot{q} = B\dot{w} + \dot{B}w. \tag{6}$$

Now let us consider a desired motion trajectory for the AUV represented in the Cartesian space. The task-space (i.e., the Cartesian space) velocity and the derivative of the generalized coordinates are related by the following relation:

$$\dot{x} = J\dot{q}, \tag{7}$$

where $x_{m\times 1}$ is the position and orientation vector in the task-space ($m \leq 6$) and $J_{m\times 6}$ is the Jacobian matrix.

In order to incorporate the desired trajectory into the dynamics of the system, we differentiate Eq. (7) which yields

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q}. \tag{8}$$

Substituting $\dot{q}$ from Eq. (5) and $\ddot{q}$ from Eq. (6) into Eq. (8), we get

$$\ddot{x} = JB\dot{w} + (J\dot{B} + \dot{J}B)w. \tag{9}$$

Finally, substituting $\dot{w}$ from Eq. (4) into the above equation, we obtain the following relationship between the task-space acceleration and the thruster forces:

$$\ddot{x} = \mu F_{\mathrm{td}} + \beta, \tag{10}$$

where $\mu_{m\times n} = JBM^{-1}E$ is the *thruster control matrix* and $\beta_{m\times 1} = (J\dot{B} + \dot{J}B)w - JBM^{-1}\zeta$ is the vector of nonlinear terms. μ, the thruster control matrix is a full-rank matrix because the thruster configuration matrix E, the inertia matrix M, the transformation matrix B, and the Jacobian matrix J are, in general, full-rank except for singular configurations (e.g., pitch singularity).

Since the thruster control matrix, μ, is a nonsquare matrix, it cannot be inverted to obtain a unique solution for the desired thruster force, F_{td}. Hence we employ Moore–Penrose generalized inverse or the pseudoinverse of μ to obtain the least-norm solution for the thruster force as follows:

$$F_{\mathrm{td}} = \mu^{+}(\ddot{x} - \beta), \tag{11}$$

where $\mu^{+} = \mu^{\mathrm{T}}(\mu\mu^{\mathrm{T}})^{-1}$ is the pseudoinverse of μ.

Eq. (11) allocates the thruster forces in a least-norm fashion for the desired task-space motion. It should be noted that the least-norm solution of the thruster force given by Eq. (11) minimizes the generalized forces in Eq. (1), because they are related by the thruster configuration matrix (Eq. (3)), which is a constant matrix.

Now we design the following control law:

$$F_{\mathrm{td}} = \mu^{+}[(\ddot{x}_{\mathrm{d}} - \beta) + K_{\mathrm{v}}(\dot{x}_{\mathrm{d}} - \dot{x}) + K_{\mathrm{p}}(x_{\mathrm{d}} - x)], \tag{12}$$

where K_{p} is the position gain matrix and K_{v} is the velocity gain matrix.

Here we have formulated the control law in the task-space (i.e., the Cartesian space). This control scheme guarantees the asymptotic reduction of the task-space error. A similar control law can be designed in the joint-space by using the pseudoinverse of the thruster configuration matrix, E. However, since the reference trajectories in most robotic applications are presented in the task-space, we choose to design the controller in the task-space. Also for a kinematically redundant system, cyclic motion in the joint-space does not necessarily imply a cyclic motion in the task-space [28] and therefore a joint-space controller may not asymptotically reduce the task-space error to zero.

Substituting F_{td} from Eq. (12) into Eq. (10) and noting that $\mu\mu^{+}=I$, where I is an identity matrix, we derive the following error equation in the task-space:

$$\ddot{e}+K_{\mathrm{v}}\dot{e}+K_{\mathrm{p}}e=0, \tag{13}$$

where $e=x_{\mathrm{d}}-x$. Thus for positive values of K_{p} and K_{v}, the error goes to zero as time goes to infinity.

2.3. Fault-tolerant decomposition

A thruster fault can occur due to various reasons. Some of the mechanical causes are: (a) breaking of the motor/propeller shaft, (b) disassembling of propeller blades, and (c) jamming or stopping of the propeller blades or the shaft. The breaking of the shaft or the disassembling of the blades will result in sudden reduction in motor load. In such a case, the motor will draw significantly less current, which can be detected easily. On the other hand, the jamming or stopping of the blade or the shaft will increase the motor load excessively and that will result in a sharp increase in current. In both cases, the thruster fault can be detected by monitoring the change in current drawn by the thrusters. Without going into the details of possible nature of thruster faults and how they can be detected, we assume in this work that we can detect thruster faults when they occur.

Here we consider the situation when one or more thrusters do not work because of faults. Our objective is to redistribute, i.e., to reallocate the thruster forces among the functioning thrusters in such a way that the AUV still follows the desired task-space trajectories. In order to incorporate the thruster faults into the mathematical formulation, we introduce the weighted pseudoinverse of the thruster control matrix, μ.

A weighted pseudoinverse is defined in the following way. Suppose $y_{m\times 1}=A_{m\times n}x_{n\times 1}$, then $x=A_{\mathrm{W}}^{+}y$, where A_{W}^{+} is the weighted pseudoinverse of A and has the expression $A_{\mathrm{W}}^{+}=W^{-1}A^{\mathrm{T}}(AW^{-1}A^{\mathrm{T}})^{-1}$. Here W is a weight matrix that can be used to adjust the contribution of the respective components of y that generates x. The derivation of the expression of weighted pseudoinverse can be found in [29].

In this case, we consider Eq. (10) to determine the weighted least-norm solution for the thruster force. We derive an expression for thruster force decomposition that minimizes the weighted norm of $\|F_{\mathrm{td}}\|_{\mathrm{W}}=(F_{\mathrm{td}}^{\mathrm{T}}WF_{\mathrm{td}})^{1/2}$, where $W_{n\times n}$ is a symmetric and positive definite weight matrix. Thus from Eq. (10), we write

$$F_{\mathrm{td}}=\mu_{\mathrm{W}}^{+}(\ddot{x}-\beta), \tag{14}$$

where $\mu_{\mathrm{W}}^{+}=W^{-1}\mu^{\mathrm{T}}(\mu W^{-1}\mu^{\mathrm{T}})^{-1}$ is the weighted pseudoinverse of μ.

We note from Eq. (14) that by changing the weights in the W matrix, the individual components of the thruster force vector can be modified. We also note that if W is a diagonal matrix and any element of the diagonal is infinity, then the corresponding thruster force will be zero. This implies that if we construct a new matrix, $\Psi=W^{-1}$, with diagonal elements either 1 or 0 then we will be able to capture the fault information of each thruster. For example, if there is no fault in any of the thrusters, then $\Psi=W^{-1}$ becomes an identity matrix and Eq. (14) becomes Eq. (11). However, when there is a fault in any thruster, we put the corresponding diagonal element of the Ψ matrix to zero, and Eq. (14) becomes different from Eq. (11). We define $\Psi_{n\times n}$ to be the *thruster fault matrix*. Rewriting Eq. (14) in terms of thruster fault matrix we obtain an expression for the allocation of thruster forces subject to thruster faults:

$$F_{\mathrm{td}}=\Psi\mu^{\mathrm{T}}(\mu\Psi\mu^{\mathrm{T}})^{-1}(\ddot{x}-\beta). \tag{15}$$

The above equation allows us to find a thruster force distribution in the presence of thruster faults. Note that this equation cannot be used when the rank of $(\mu\Psi\mu^{\mathrm{T}})$ becomes less than m. Since μ is assumed to be full-rank (as discussed before), the only way it can happen is when the thruster fault matrix, Ψ, which is an $n\times n$ matrix, reduces

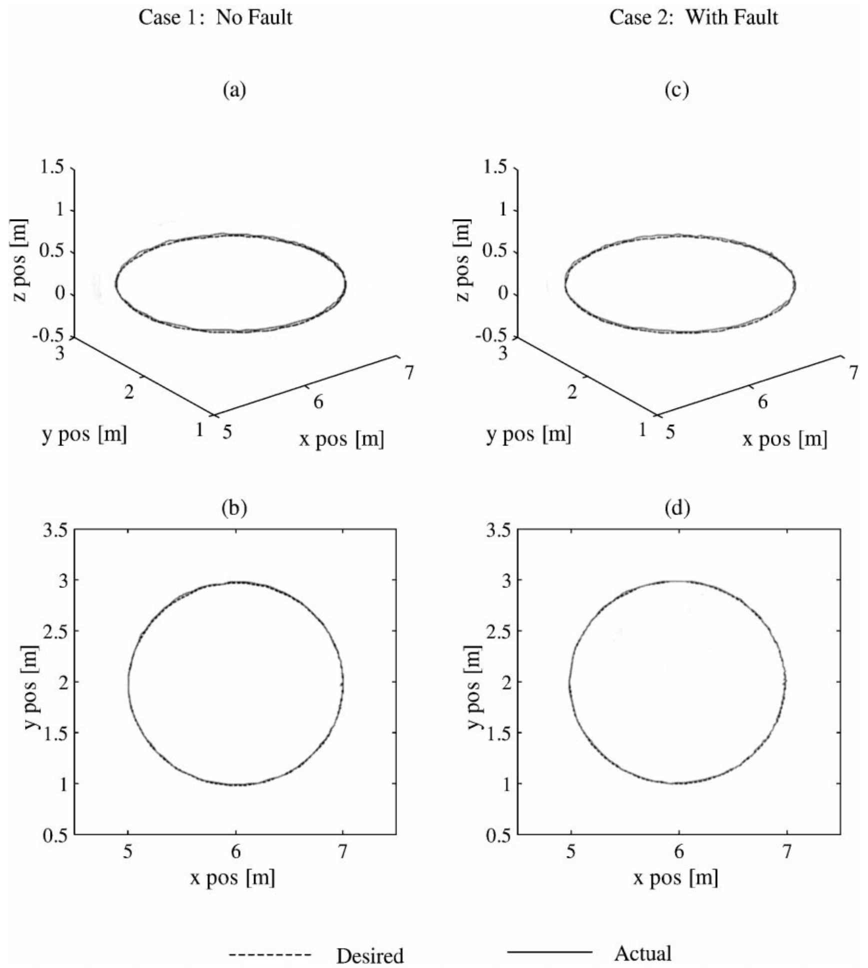


Fig. 1. Desired and actual task-space (Cartesian) paths: (a) and (b) no fault; (c) and (d) with faults.

the rank of $(\mu\Psi\mu^{\mathrm{T}})$ by more than $n-m$. This can happen when there are too many faulty thrusters resulting in such a situation that the thruster forces no longer span the six-dimensional task-space.

It should also be mentioned that in the above formulation, we have assumed that thruster faults are detected by using any standard technique. This information about thruster faults is utilized in designing the thruster fault matrix, Ψ.

We define a similar control law as given by Eq. (12) for this case to obtain

$$F_{\mathrm{td}} = \Psi\mu^{\mathrm{T}}(\mu\Psi\mu^{\mathrm{T}})^{-1}[(\ddot{x}_{\mathrm{d}} - \beta) + K_{\mathrm{v}}(\dot{x}_{\mathrm{d}} - x) + K_{\mathrm{p}}(x_{\mathrm{d}} - x)]. \tag{16}$$

When F_{td} as obtained from Eq. (16) is applied to Eq. (10), we obtain a similar error equation as given by Eq. (13).

We note here that the energy of the system is the product of the generalized forces and the corresponding displacements. In this work, the thruster forces are allocated in a least-norm fashion. Such a least-norm distribution of the thruster forces minimizes the generalized forces. Since the controller reduces the position errors to

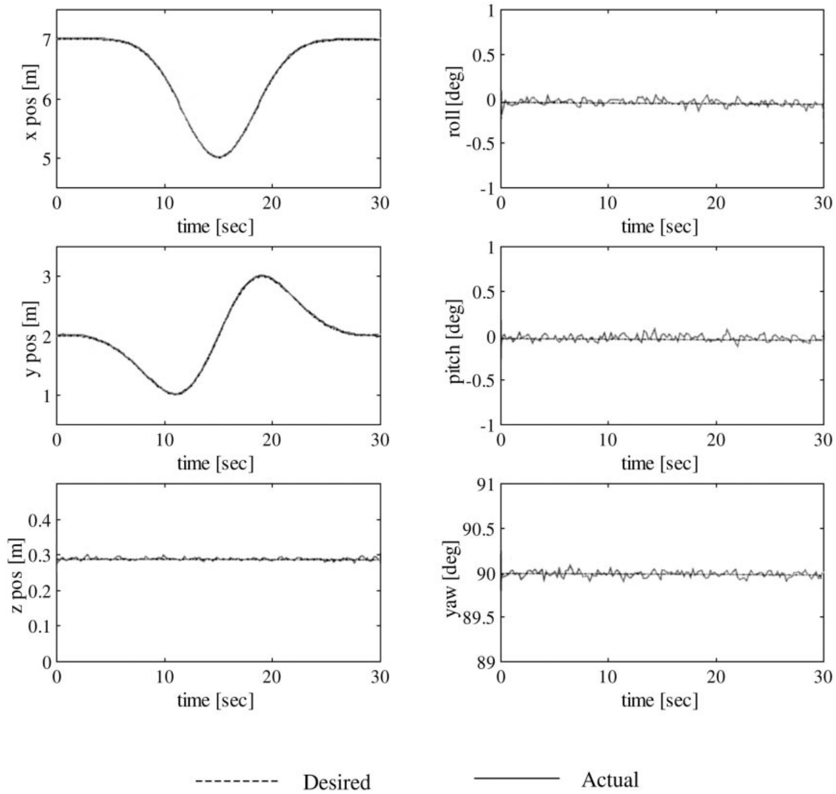


Fig. 2. Task-space trajectories for Case 1 (no fault).

zero, it implies that the minimization of the generalized forces leads to the minimization of the energy of the system.

3. Results and discussion

We have performed computer simulations to determine the efficacy of our proposed control scheme. We used Omni Directional Intelligent Navigator (ODIN), which was designed in the University of Hawaii [26], as our model. ODIN is a near-spherical AUV, which has four horizontal thrusters and four vertical thrusters.

3.1. Simulation model

The dynamic model of ODIN is given by

$$[M_{\mathrm{RB}} + M_{\mathrm{A}}]\dot{w} + [C_{\mathrm{RB}} + C_{\mathrm{A}}]w + Dw + G = \tau, \tag{17}$$

where RB represents the rigid body and A represents the added mass terms of the relevant parameters. The expressions for the relevant parameters are given by

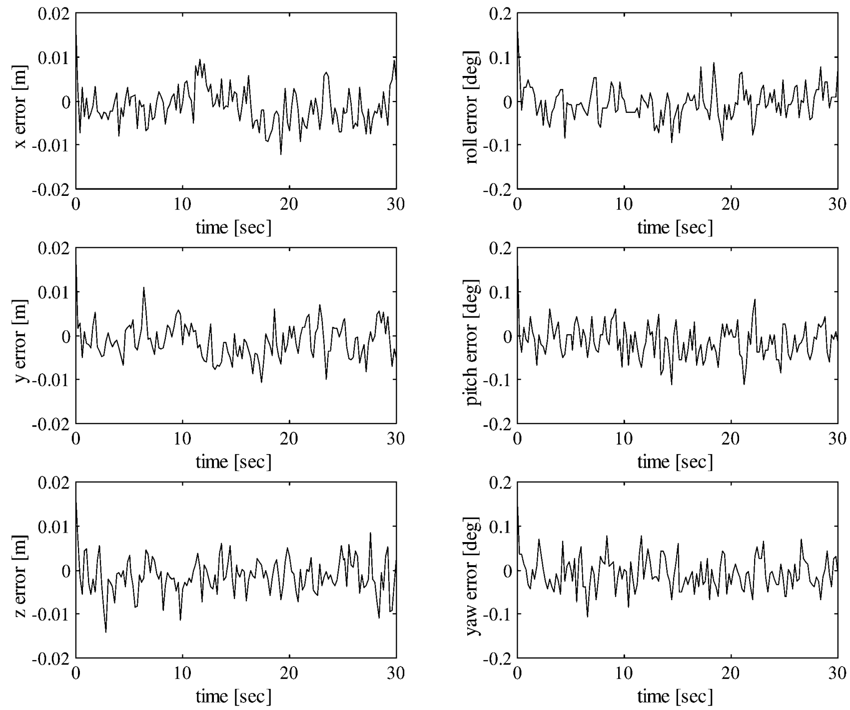


Fig. 3. Task-space trajectory errors for Case 1 (no fault).

$$M_{\mathrm{RB}} = \begin{bmatrix} m & 0 & 0 & 0 & mz_{\mathrm{G}} & 0 \\ 0 & m & 0 & -mz_{\mathrm{G}} & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & -mz_{\mathrm{G}} & 0 & I_{xx} & 0 & 0 \\ mz_{\mathrm{G}} & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix},$$

where $I_{xx} = I_{yy} = I_{zz} = I = \frac{8}{15}\pi\rho_{\mathrm{v}}r^5$ are the moments of inertia about the principal axes.

$$M_{\mathrm{A}} = \begin{bmatrix} \frac{2}{3}\pi\rho r^3 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{2}{3}\pi\rho r^3 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{3}\pi\rho r^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

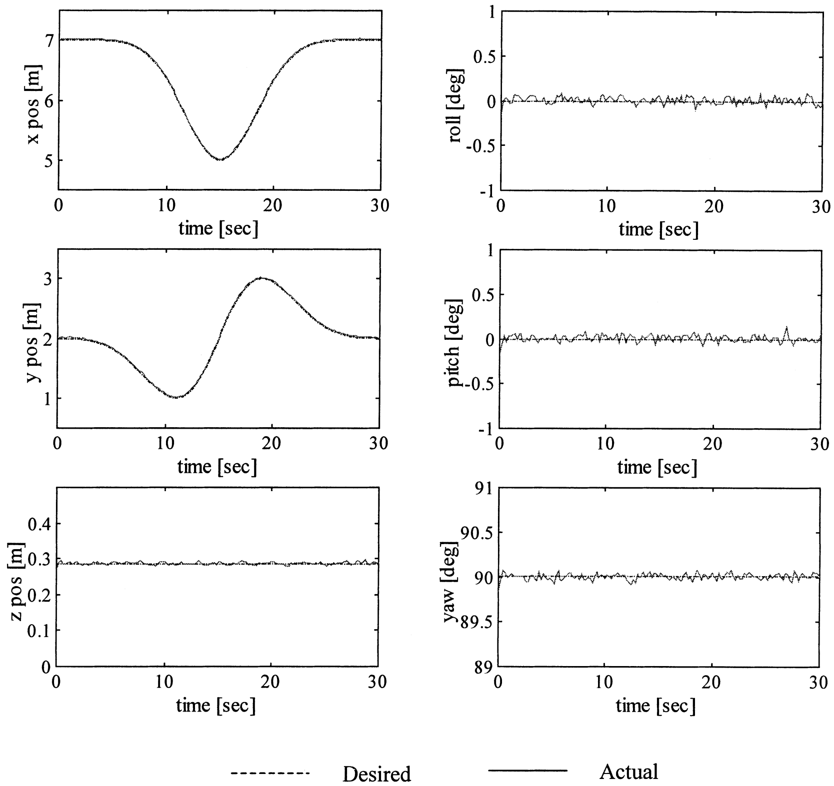


Fig. 4. Task-space trajectories for Case 2 (with faults).

$$
C_{\mathrm{RB}} = \begin{bmatrix} 0 & 0 & 0 & mz_{\mathrm{G}}w_6 & mw_3 & -mw_2 \\ 0 & 0 & 0 & -mw_3 & mz_{\mathrm{G}}w_6 & mw_1 \\ 0 & 0 & 0 & m(w_2 - z_{\mathrm{G}}w_4) & -m(w_1 + z_{\mathrm{G}}w_5) & 0 \\ -mz_{\mathrm{G}}w_6 & mw_3 & -m(w_2 - z_{\mathrm{G}}w_4) & 0 & Iw_6 & -Iw_5 \\ -mw_3 & -mz_{\mathrm{G}}w_6 & m(w_1 + z_{\mathrm{G}}w_5) & -Iw_6 & 0 & Iw_4 \\ mw_2 & -mw_1 & 0 & Iw_5 & -Iw_4 & 0 \end{bmatrix},
$$

$$
C_{\mathrm{A}} = \begin{bmatrix} 0 & 0 & 0 & 0 & mw_3 & -mw_2 \\ 0 & 0 & 0 & -mw_3 & 0 & mw_1 \\ 0 & 0 & 0 & mw_2 & -mw_1 & 0 \\ 0 & mw_3 & -mw_2 & 0 & 0 & 0 \\ -mw_3 & 0 & mw_1 & 0 & 0 & 0 \\ mw_2 & -mw_1 & 0 & 0 & 0 & 0 \end{bmatrix},
$$

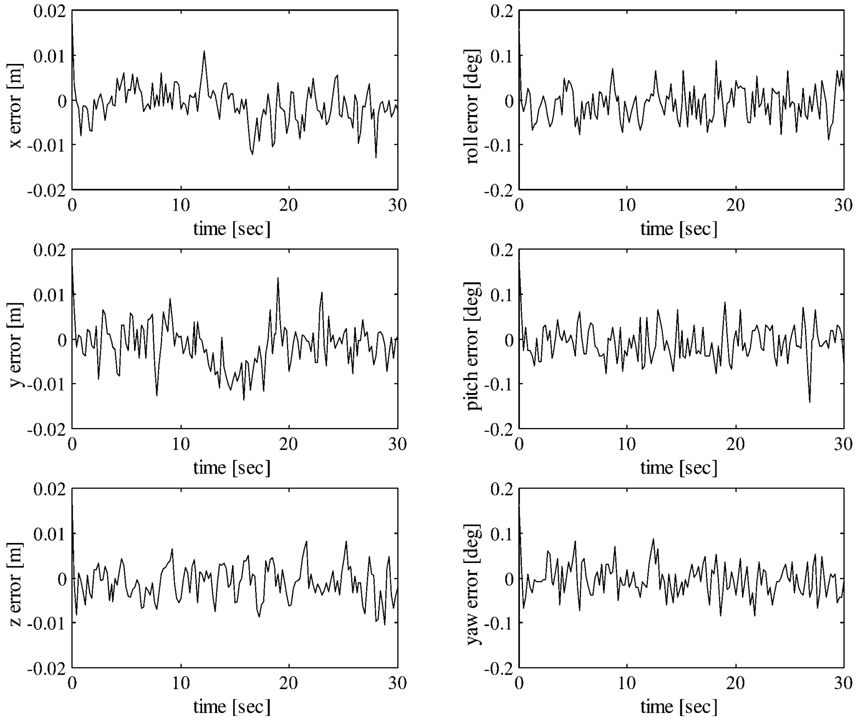


Fig. 5. Task-space trajectory errors for Case 2 (with faults).

$$D=\begin{bmatrix} d_{t1}\,|w_1|+d_{t2} & 0 & 0 & 0 & 0 & 0\\ 0 & d_{t1}\,|w_2|+d_{t2} & 0 & 0 & 0 & 0\\ 0 & 0 & d_{t1}|w_3|+d_{t2} & 0 & 0 & 0\\ 0 & 0 & 0 & d_1|w_4|+d_2 & 0 & 0\\ 0 & 0 & 0 & 0 & d_1|w_5|+d_2 & 0\\ 0 & 0 & 0 & 0 & 0 & d_1|w_6|+d_2 \end{bmatrix},$$

$$G=\begin{bmatrix} (mg-\frac{4}{3}\pi r^3\rho g)\sin(q_5)\\ -(mg-\frac{4}{3}\pi r^3\rho g)\cos(q_5)\sin(q_4)\\ -(mg-\frac{4}{3}\pi r^3\rho g)\cos(q_5)\cos(q_4)\\ z_G mg\cos(q_5)\sin(q_4)\\ z_G mg\sin(q_5)\\ 0 \end{bmatrix}.$$

Here $r=0.31$ m is the radius of ODIN, $m=125.0$ kg the mass of ODIN, $z_G=0.05$ m the distance of the CG from the geometric center, $\rho=1000\,\mathrm{kg/m^3}$ the density of water, $\rho_v=965\,\mathrm{kg/m^3}$ the average density of the ODIN AUV, $g=9.81\,\mathrm{m/s^2}$, $d_{t1}=148\,\mathrm{N(s/m)^2}$ the translational quadratic damping factor, $d_{t2}=100\,\mathrm{N(s/m)^2}$ the translational

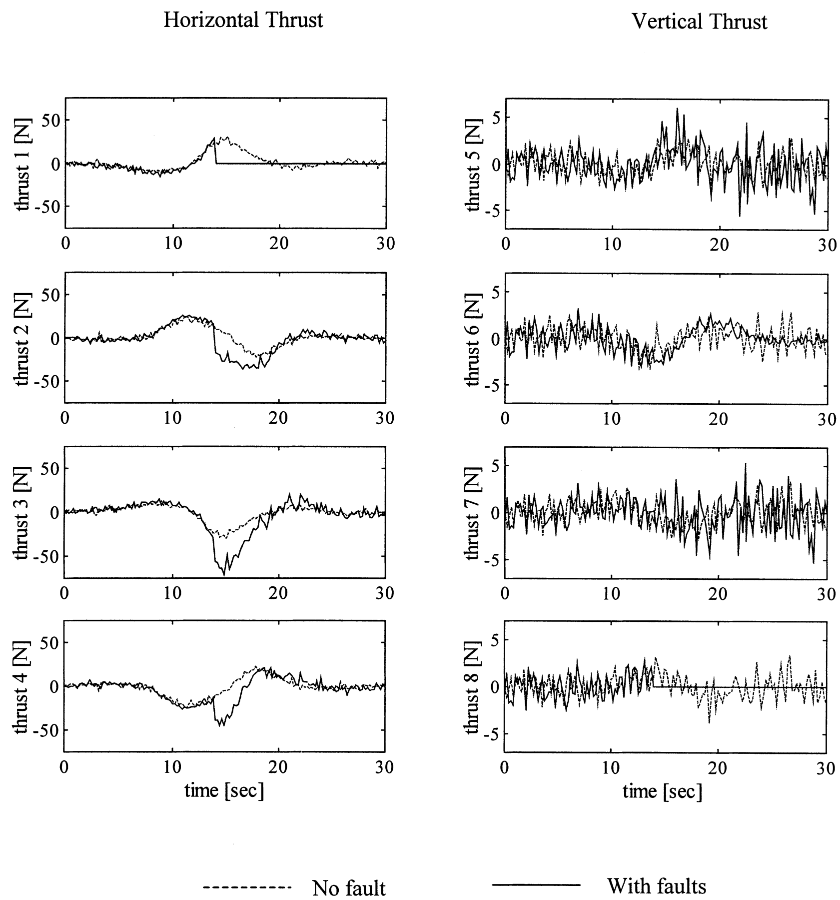


Fig. 6. Thruster forces for both Case 1 (no fault) and Case 2 (with faults), thrusts 1–4 for horizontal thrusters, and thrusts 5–8 for vertical thrusters.

linear damping factor, $d_1 = 280\,\mathrm{N\,s^2/m}$ the angular quadratic damping factor, and $d_2 = 230\,\mathrm{N\,s^2/m}$ is the angular linear damping factor.

The thruster configuration matrix for ODIN is

$$E = \begin{bmatrix} s & -s & -s & s & 0 & 0 & 0 & 0 \\ s & s & -s & -s & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & R\,s & R\,s & -R\,s & -R\,s \\ 0 & 0 & 0 & 0 & R\,s & -R\,s & -R\,s & R\,s \\ R_z & -R_z & R_z & -R_z & 0 & 0 & 0 & 0 \end{bmatrix},$$

where $s = \sin(\frac{1}{4}\pi)$, $R = 0.381$ m is the distance from the center of the vehicle to the vertical thruster's center, and $R_z = 0.508$ m is the radial distance from the center of the vehicle to the horizontal thruster's center.

3.2. Results and discussion

We present simulation results for two cases to demonstrate the effectiveness of the proposed method. In Case 1, all the thrusters are in working condition and therefore the thruster fault matrix, Ψ, is an identity matrix. In this case, ODIN tries to track a circular path of diameter 2.0 m in horizontal plane in 30.0 s. The vehicle attitudes are kept constant at $[0^\circ, 0^\circ, 90^\circ]$. The model is assumed to be accurate. But we have incorporated Gaussian noise in our simulation. The noise has 1.0 mm mean and 1.5 mm standard deviation for linear position measurements, and 0.1° mean and 0.1° standard deviation for angular orientation measurements. In Case 2, ODIN tries to track the same circular path as before, however, in this case, one horizontal thruster (Thruster 1) and one vertical thruster (Thruster 8) stop functioning after 12.0 s of the motion. Additionally, in Case 2, we assumed an inaccurate model where the hydrodynamic terms M_{A}, C_{A} and D are estimated 25% more than the actual values. Even in this case, in spite of the thruster faults, the control law given by Eq. (16) enables ODIN to follow the circular path quite accurately, although the thruster force allocation was different from that of Case 1.

We present the simulation results in Figs. 1–6. From Fig. 1, it is observed that in both the cases, the AUV follows the geometric path quite accurately. In Figs. 2 and 3, we have plotted the task-space trajectories and the task-space position errors for Case 1. We have plotted the task-space trajectories and task-space position errors in Figs. 4 and 5 for Case 2. It can be seen that the trajectory tracking errors for both the cases are small. It is encouraging to observe that the controller performs well in the presence of sensory noise (for both Cases 1 and 2) and modeling uncertainty (25% uncertainty for Case 2). The thruster forces for both the cases are plotted in Fig. 6. We have simulated two thruster faults: one is a horizontal thruster (Thruster 1, shown in the first subplot) and the other is a vertical thruster (Thruster 8, shown in the last subplot). These thrusters are mounted on the same bracket of the ODIN. We considered this case because it was one of the worst thruster faults that can arise during operation. From Fig. 6, it is seen that in case of thruster faults, the faulty thrusters supply zero force. But the other working thrusters reallocate the remaining thruster forces among themselves so that the desired task-space trajectories are tracked accurately. In all these simulation examples, the thruster forces are within the limits of the actual thrusters of the ODIN AUV.

4. Conclusions

We have presented a new approach to the allocation of the thruster forces of an AUV. The allocation technique provides the minimum norm solution to the thruster forces for a particular task-space motion trajectory. We have further extended this allocation technique to include fault information of each thruster. Thus the proposed scheme automatically allocates the thruster force for a given trajectory based on thruster fault information. We have also proposed a task-space control law that allows the AUV to track the desired trajectory with asymptotic reduction of error. Results from computer simulations demonstrate the efficacy of the proposed technique.

This proposed thruster force allocation technique is not without its limitations. One important limitation of this scheme lies in its inability to deal with the thruster force saturation problem. In such a case, it cannot reallocate the thruster forces without degrading the performance of the controller. It will not be possible to control forces with the present formulation because in such a case, the output of and the input to the controller will be *algebraically* related. Currently we are working on the extension of this proposed framework to handle the thruster force saturation problem. Future work will also include further refinement of the model and experimental implementation of the proposed scheme on the ODIN AUV.

Acknowledgements

This work was partially supported by the National Science Foundation under grant BES97-01614 and the Office of Naval Research under grant N00014-97-1-0961.

References

[1] D.R. Yoerger, J.J.E. Slotine, Robust trajectory control of underwater vehicles, IEEE Journal of Oceanic Engineering 10 (4) (1990) 462–470.

[2] K.R. Goheen, R.E. Jeffery, Multivariable self-tuning autopilots for autonomous and remotely operated underwater vehicles, IEEE Journal of Oceanic Engineering 15 (3) (1990) 144–151.

[3] Y. Nakamura, S. Savant, Nonholonomic motion control of an autonomous underwater vehicle, in: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Osaka, Japan, 1991, pp. 1254–1259.

[4] Y. Nakamura, S. Savant, Nonlinear tracking control of autonomous underwater vehicles, in: Proceedings of the IEEE Conference on Robotics and Automation, Nice, France, 1992, pp. A4–A9.

[5] A.J. Healey, D.B. Marco, Slow speed flight control of autonomous underwater vehicles: experimental results with NPS AUV II, in: Proceedings of the Second International Offshore and Polar Engineering Conference, San Francisco, CA, June 1992, pp. 523–532.

[6] A.J. Healey, D. Lienard, Multivariable sliding-mode control of autonomous underwater vehicles, IEEE Journal of Oceanic Engineering 15 (3) (1993) 161–166.

[7] T.I. Fossen, Underwater vehicle dynamics, in: J. Yuh (Ed.), Underwater Robotic Vehicles: Design and Control, TSI Press, 1995, pp. 15–40.

[8] N. Sarkar, J. Yuh, T.K. Podder, Adaptive control of underwater vehicle-manipulator systems subject to joint limits, in: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Kyongju, Korea, 1999, pp. 142–147.

[9] J. Yuh, Learning control of underwater robotic vehicles, IEEE Control System Magazine 15 (2) (1994) 39–46.

[10] K. Ishii, T. Fujii, T. Ura, Neural network system for on-line controller adaptation and its application to underwater robot, in: Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, 1998, pp. 756–761.

[11] T.W. McLain, R.W. Beard, Successive Galerkin approximation to the nonlinear optimal control of an underwater robotic vehicle, in: Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, 1998, pp. 762–767.

[12] D. Yoerger, J. Cooke, J.J. Slotine, The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design, IEEE Journal of Oceanic Engineering 15 (3) (1990) 167–178.

[13] A.J. Healey, S.M. Rock, S. Cody, D. Miles, J.P. Brown, Toward an improved understanding of thruster dynamics for underwater vehicles, IEEE Journal of Oceanic Engineering 20 (4) (1995) 354–361.

[14] L.L. Whitcomb, D.R. Yoerger, Preliminary experiments in the model-based dynamic control of marine thrusters, in: Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, MN, April 1996, pp. 2166–2173.

[15] M.M. Moya, W.M. Davidson, Sensor-driven fault control of a maintenance robot, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, April 1986, pp. 428–434.

[16] A. Orrick, M. McDermott, Fault detection of an autonomous vehicle, in: Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology, 1994, pp. 377–382.

[17] J.J. Gertler, K.C. Anderson, An evidential reasoning extension to quantitative model-based failure diagnosis, IEEE Transactions on Systems, Man, and Cybernetics 22 (2) (1992) 275–289.

[18] P.M. Frank, Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results, Automatica 26 (3) (1990) 457–474.

[19] M. Takai, T. Fujii, T. Ura, A model based diagnosis system for autonomous underwater vehicles using artificial neural networks, in: Proceedings of the Ninth International Symposium on Unmanned Untethered Submersible Technology, Durham, NH, September 1995, pp. 243–252.

[20] J.V. Medanic, W.R. Perkins, R.J. Veillette, Design of reliable control systems, IEEE Transactions on Automatic Control 37 (3) (1992) 290–304.

[21] M.M. Polycarpou, A.T. Vemuri, Learning methodology for failure detection and accommodation, IEEE Control Systems Magazine 15 (3) (1995) 16–24.

[22] A.J. Healey, A neural network approach to failure diagnosis for underwater vehicles, in: Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology, June 1992, pp. 131–134.

[23] E.G. Laukonen, K.M. Passino, V. Krishnaswami, G.C. Luh, G. Rizzoni, Fault detection and isolation for an experimental internal combustion engine via fuzzy identification, IEEE Transactions on Control Systems Technology 3 (3) (1995) 347–355.

[24] T.A. Healy, L.J. Kerr, L.J. Larkin, Model based fuzzy logic sensor fault accommodation, Journal of Engineering for Gas Turbines and Power 120 (3) (1998) 533–536.

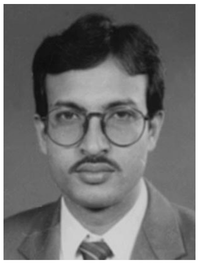
[25] H.E. Rauch, Intelligent fault diagnosis and control reconfiguration, IEEE Control Systems Magazine 14 (3) (1994) 6–12.

[26] K.C. Yang, J. Yuh, S.K. Choi, Experimental study of fault tolerance system design for underwater robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, May 1998, pp. 1051–1056.

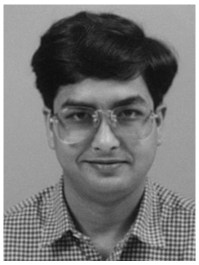
[27] T.I. Fossen, Guidance and Control of Ocean Vehicles, Wiley, New York, 1994.

[28] C.A. Klein, C.S. Huang, Review of pseudoinverse control for use with kinematically redundant manipulators, IEEE Transactions on Systems, Man, and Cybernetics SMC-13 (2) (1983) 245–250.

[29] A. Ben-Israel, T.N.E. Greville, Generalized Inverse: Theory and Applications, Wiley, New York, 1974.



Tarun Kanti Podder received his B.E. degree in Mechanical Engineering from the University of Calcutta, India, in 1988, and M.E. from the Indian Institute of Science, Bangalore, India, in 1990. Since 1997, he has been pursuing doctoral research in the Department of Mechanical Engineering, University of Hawaii at Manoa, Honolulu, USA. His research area covers dynamics, path planning and control of underwater robotic system. From 1990 to 1991, he worked as a Research Engineer in the Robotics Center, Indian Institute of Technology, Kanpur, India. He was a lecturer in the Department of Mechanical Engineering, B.E. College, University of Calcutta, from 1991 to 1997. He is a life member of the Institution of Engineers (India), life member of the All India Council for Technical Education, and member of IEEE.



Nilanjan Sarkar is an Assistant Professor in the Mechanical Engineering Department of Vanderbilt University, Nashville, TN, USA. He was an Assistant Professor at the University of Hawaii at Manoa prior to joining Vanderbilt University. He received his Ph.D. degree in Mechanical Engineering and Applied Mechanics from the University of Pennsylvania in 1993 and subsequently joined Queen's University, Canada as a Postdoctoral Fellow in the Computing and Information Science Department. Professor Sarkar's research interests lie in dynamics, control and planning of autonomous systems. He heads the Autonomous Systems Laboratory and is a member of the Center of Intelligent Systems at Vanderbilt University. He teaches kinematics, dynamics, control, and robotic courses. He is a member of IEEE and ASME.