Fuzzy Redundancy Resolution and Motion Coordination for Underwater Vehicle-Manipulator Systems

Gianluca Antonelli, *Member, IEEE,* and Stefano Chiaverini, *Member, IEEE*

***Abstract*—The problem of redundancy resolution and motion coordination between the vehicle and the manipulator in underwater vehicle-manipulator systems (UVMSs) is addressed in this paper. UVMSs usually possess more degrees of freedom than those required to perform end-effector tasks; therefore, they are redundant systems and kinematic control techniques can be applied aimed at achieving additional control objectives besides tracking of the end-effector trajectory. In this paper, a task-priority inverse kinematics approach to redundancy resolution is merged with a fuzzy technique to manage the vehicle-arm coordination. The fuzzy technique is used both to distribute the motion between vehicle and manipulator and to handle multiple secondary tasks. Numerical case studies are developed to demonstrate effectiveness of the proposed technique.**

***Index Terms*—Kinematic control, redundancy resolution, underwater vehicle-manipulator systems.**

I. Introduction

A ROBOTIC system is kinematically redundant when it possesses more degrees-of-freedom (DOFs) than those required to execute a given task. A generic manipulation task is usually given in terms of position/orientation trajectories for the end effector. In this sense, an underwater vehicle-manipulator system (UVMS) is always kinematically redundant due to the DOFs provided by the vehicle itself. However, it is not always efficient to use vehicle thrusters to move the manipulator end effector because of the difficulty of controlling the vehicle in hovering. Moreover, due to the different inertia between vehicle and manipulator, movement of the latter is energetically more efficient. On the other hand, reconfiguration of the whole system is required when the manipulator is working at the boundaries of its workspace or close to a kinematic singularity; motion of the sole manipulator, thus, is not always possible or efficient. Also, offline trajectory planning is not always possible in unstructured environments as in case of underwater autonomous missions.

When a manipulation task has to be performed with an UVMS, the system is usually kept in a confined space (e.g., underwater structure maintenance). The vehicle is then used to ensure station keeping. However, motion of the vehicle can be required to specific purposes, e.g., inspection of a pipeline, reconfiguration of the system, real-time motion coordination while performing end-effector trajectory tracking.

According to this, a redundancy resolution technique might be useful to achieve system coordination in such a way as to guarantee end-effector tracking accuracy and, at the same time, additional control objectives, e.g., energy savings or increase of system manipulability. To this purpose, the task-priority redundancy resolution technique [14], [17] is well suited in that it allows the specification of a primary task which is fulfilled with higher priority with respect to a secondary task.

Control of end-effector position/orientation can be obtained also with dynamic control by suitably expressing the mathematical model [21], [23]. This approach, successfully implemented for industrial robots, is not suitable for UVMS's for two main reasons: First, in underwater environment the dynamic parameters are usually poorly known. Second, the redundancy of the system is not exploited.

Limiting our attention to UVMSs, few papers have addressed the problem of inverse kinematics resolution. In [4], a task priority approach has been proposed aimed at fulfilling secondary tasks such as reduction of fuel consumption, improvement of system manipulability, and obstacle avoidance. In [19], a second-order inverse kinematics approach is developed to reduce the drag forces of the system; to the purpose, however, full dynamic compensation, and thus knowledge of the dynamic parameters, is required.

In this paper the task-priority redundancy resolution technique for kinematic control of UVMSs presented in [4] is integrated with a fuzzy approach preliminary investigated in [2], [3]. A fuzzy inference system (FIS) is in charge of distributing the required end-effector motion between the vehicle and the manipulator. At the same time, the FIS can activate a secondary task if the corresponding variable is out of a safe range. Notice that several secondary tasks can be defined with this approach, moreover, a suitable use of the fuzzy concepts allows the flexible handling of a large number of variable of interest. The proposed inverse kinematic approach is based on the work in [8] and, thus, is robust to the occurrence of algorithmic singularities.

Numerical simulations have been developed on a UVMS constituted by a 6-DOFs vehicle carrying a 6-DOF; the overall system, thus, has 12-DOFs. The obtained results show the advantage of the proposed approach.

Manuscript received February 26, 2002; revised June 3, 2002 and July 5, 2002.

The authors are with the Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale, Università degli Studi di Cassino, 03043 Cassino, Italy (e-mail: antonelli@unicas.it; chiaverini@unicas.it).

Digital Object Identifier 10.1109/TFUZZ.2002.806321

II. Kinematics

The vehicle is completely described by its position and orientation with respect to a reference frame that we will suppose earth-fixed and inertial. Let define the vector $\boldsymbol{\eta} = [\boldsymbol{\eta}_1^T \quad \boldsymbol{\eta}_2^T]^T$, where $\boldsymbol{\eta}_1 = [x \quad y \quad z]^T \in \mathbb{R}^3$ is the vector of vehicle position coordinates in a earth-fixed reference frame, and $\boldsymbol{\eta}_2 = [\phi \quad \theta \quad \psi]^T \in \mathbb{R}^3$ is the vector of vehicle Roll-Pitch-Yaw Euler-angle coordinates in a earth-fixed reference frame, corresponding to the ZYX Euler-angle type. The vectors $\dot{\boldsymbol{\eta}}_1$, $\dot{\boldsymbol{\eta}}_2$ are the corresponding time derivatives (expressed in the earth-fixed frame).

It is useful to define the vehicle's velocity in a vehicle-fixed frame; let $\boldsymbol{\nu} = [\boldsymbol{\nu}_1^T \quad \boldsymbol{\nu}_2^T]^T$, where $\boldsymbol{\nu}_1 = [u \quad v \quad \omega]^T \in \mathbb{R}^3$ is the linear velocity of the vehicle-fixed frame with respect to the earth-fixed frame expressed in the vehicle-fixed frame and $\boldsymbol{\nu}_2 = [p \quad q \quad r]^T \in \mathbb{R}^3$ is the angular velocity of the vehicle-fixed frame with respect to the earth-fixed frame expressed in the vehicle-fixed frame. The defined velocity vector satisfy the following equations:

$$\boldsymbol{\nu}_1 = \boldsymbol{R}_I^B \dot{\boldsymbol{\eta}}_1 \tag{1}$$

$$\boldsymbol{\nu}_2 = \boldsymbol{T}(\boldsymbol{\eta}_2)\dot{\boldsymbol{\eta}}_2 \tag{2}$$

where $\boldsymbol{R}_I^B$ is the rotation matrix expressing the transformation from the earth-fixed frame to the vehicle-fixed frame and $\boldsymbol{T}$ is the matrix

$$\boldsymbol{T}(\boldsymbol{\eta}_2) = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 0 & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix}. \tag{3}$$

Let us also define $\boldsymbol{q} = [q_1 \cdots q_n]^T \in \mathbb{R}^n$ as the vector of joint positions, n being the number of joints, and $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ as the corresponding time derivative.

By introducing the vector $\boldsymbol{\zeta} = [\boldsymbol{\nu}_1^T \quad \boldsymbol{\nu}_2^T \quad \dot{\boldsymbol{q}}^T]^T$ it is possible to rewrite the relation between the above velocities in compact form:

$$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{R}_I^B & \boldsymbol{O}_{3\times 3} & \boldsymbol{O}_{3\times n} \\ \boldsymbol{O}_{3\times 3} & \boldsymbol{T} & \boldsymbol{O}_{3\times n} \\ \boldsymbol{O}_{n\times 3} & \boldsymbol{O}_{n\times 3} & \boldsymbol{I}_n \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\eta}} \\ \dot{\boldsymbol{q}} \end{bmatrix} = \boldsymbol{J}_k \begin{bmatrix} \dot{\boldsymbol{\eta}} \\ \dot{\boldsymbol{q}} \end{bmatrix} \tag{4}$$

where $\boldsymbol{I}_n$ is the $(n \times n)$ identity matrix and $\boldsymbol{O}_{n_1 \times n_2}$ is the $(n_1 \times n_2)$ null matrix.

Since the task to be executed requires position/orientation control of the end effector frame, it is necessary to consider the position/orientation of the end effector in the earth-fixed frame, $\boldsymbol{\eta}_{ee} = [\boldsymbol{\eta}_{ee1}^T \quad \boldsymbol{\eta}_{ee2}^T]^T$, where $\boldsymbol{\eta}_{ee1} \in \mathbb{R}^3$ is the end-effector position expressed in the earth-fixed frame and $\boldsymbol{\eta}_{ee2} \in \mathbb{R}^3$ is the orientation of the end effector in the earth-fixed frame expressed by Euler angles; the vector $\boldsymbol{\eta}_{ee1}$ is a function of the system configuration, i.e., $\boldsymbol{\eta}_{ee1}(\boldsymbol{\eta}, \boldsymbol{q})$. The vector $\boldsymbol{\eta}_{ee2}$ is a function of the system configuration too, but does not depend from the vehicle position, i.e., $\boldsymbol{\eta}_{ee2}(\boldsymbol{\eta}_2, \boldsymbol{q})$. The vector $\dot{\boldsymbol{\eta}}_{ee1} \in \mathbb{R}^3$ and $\dot{\boldsymbol{\eta}}_{ee2} \in \mathbb{R}^3$ are the corresponding time derivative. The relation between the end-effector posture and the system configuration can be expressed by the following nonlinear equation:

$$\begin{bmatrix} \boldsymbol{\eta}_{ee1} \\ \boldsymbol{\eta}_{ee2} \end{bmatrix} = \boldsymbol{k}(\boldsymbol{\eta}, \boldsymbol{q}). \tag{5}$$

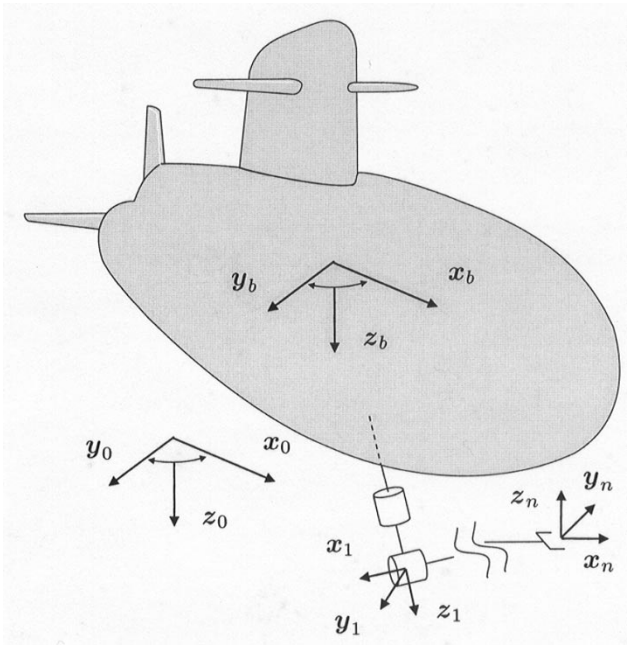

Fig. 1. Sketch of an UVMS with relevant frames.

The vectors $\dot{\boldsymbol{\eta}}_{ee1}$ and $\dot{\boldsymbol{\eta}}_{ee2}$ are related to the body-fixed velocities $\boldsymbol{\nu}_{ee}$ via relations analogous to (1) and (2), namely

$$\boldsymbol{\nu}_{ee1} = \boldsymbol{R}_I^n \dot{\boldsymbol{\eta}}_{ee1} \tag{6}$$

$$\boldsymbol{\nu}_{ee2} = \boldsymbol{T}(\boldsymbol{\eta}_{ee2})\dot{\boldsymbol{\eta}}_{ee2} \tag{7}$$

where $\boldsymbol{R}_I^n$ is the rotation matrix from the earth-fixed frame to the end-effector frame (i.e., frame n, see Fig. 1) and $\boldsymbol{T}$ is the matrix defined in (3) computed with the use of the Euler angles of the end-effector frame.

The end-effector velocities (expressed in the earth-fixed frame) are related to the body-fixed system velocity by a suitable Jacobian matrix, i.e.,

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}_{ee1} \\ \boldsymbol{R}_n^I \boldsymbol{\nu}_{ee2} \end{bmatrix} = \boldsymbol{J}(\boldsymbol{R}_B^I, \boldsymbol{q})\boldsymbol{\zeta}. \tag{8}$$

Notice that the Jacobian has been derived with respect to the angular velocity of the end effector expressed in the earth-fixed frame (the matrix $\boldsymbol{R}_n^I = \boldsymbol{R}_I^{n^T}$ is the rotation from the frame n the the earth-fixed frame). This is done in order to implement a closed-loop inverse kinematics expressed by quaternions as will be shown in Section III.

III. Kinematic Control

A manipulation task is usually given in terms of position and orientation trajectory of the end effector. The objective of kinematic control is to find suitable vehicle/joint trajectories $\boldsymbol{\eta}(t)$, $\boldsymbol{q}(t)$ that correspond to a desired end-effector trajectory $\boldsymbol{\eta}_{ee,d}(t)$. The output of the inverse kinematics algorithm $\boldsymbol{\eta}_r(t)$, $\boldsymbol{q}_r(t)$ provides the reference values to the control law of the UVMS. This control law will be in charge of computing the driving forces aimed at tracking the reference trajectory for the system while counteracting dynamic effects, external disturbances, and modeling errors.

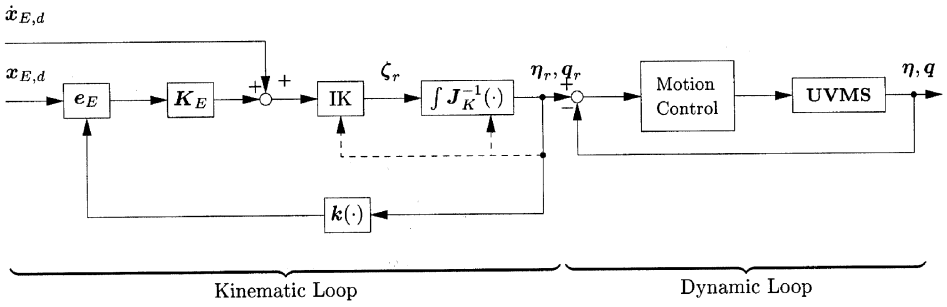

Fig. 2. Kinematic and dynamic loops. The block labeled e_E is defined by (19).

Equation (5) is invertible only for specific kinematic structures with fixed base. Moreover, the complexity of the relation and the number of solutions, i.e., different joint configurations that correspond to the same end-effector posture, increase with the degrees of freedom. As an example, a simple two-link planar manipulator with fixed base admits two different solutions for a given end-effector position while up to 16 solutions can be found in the case of 6-DOFs structures. At differential level, however, the relation between joints and end-effector velocities is much more tractable and a theory of kinematic control has been established aimed at solving inverse kinematics of generic kinematic structure.

Equation (8) maps the $(6+n)$-dimensional vehicle/joint velocities into the m-dimensional end-effector task velocities. If the UVMS has more degrees of freedom than those required to execute a given task, i.e., if $(6+n) > m$, the system is redundant with respect to the specific task and the (5)–(8) admit infinite solutions. Kinematic redundancy can be exploited to achieve additional task, beside the given end-effector task. In the following, the typical case $(6+n) \geq m$ will be considered.

The configurations at which $\boldsymbol{J}$ is rank deficient, i.e., $\mathrm{rank}(\boldsymbol{J}) < m$, are termed kinematic singularities. Kinematic singularities are of great interest for several reasons, at a singularity, in fact, the follwoing are true.

- The mobility of the structure is reduced. If the manipulator is not redundant, this implies that it is not possible to give an arbitrary motion to the end effector.
- Infinite solutions to the inverse kinematics problem may exist.
- *Close* to a kinematic singularity at small task velocities can correspond large joint velocities.

Notice that, in case of UVMS, the Jacobian has always full rank due to the mobility of the vehicle, i.e., a rigid body with 6-DOFs. However, as it will be shown in next Sections, movement of the vehicle has to be avoided when unnecessary.

A. Pseudoinverse

The simplest way to invert the mapping (8) is to use the pseudoinverse of the Jacobian matrix [24]

$$\boldsymbol{\zeta}_r = \boldsymbol{J}^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\dot{\boldsymbol{x}}_{E,d} \tag{9}$$

where $\dot{\boldsymbol{x}}_{E,d}$ is the end-effector task defined in (8) and

$$\boldsymbol{J}^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q}) = \boldsymbol{J}^{T}(\boldsymbol{\eta}, \boldsymbol{q}) \left(\boldsymbol{J}(\boldsymbol{\eta}, \boldsymbol{q})\boldsymbol{J}^{T}(\boldsymbol{\eta}, \boldsymbol{q})\right)^{-1}. \tag{10}$$

This solution corresponds to the minimization of the vehicle/joint velocities in a least-square sense [20]. Notice that subscript r in $\boldsymbol{\zeta}_r$ stands for *reference value*, meaning that those velocities are the desired value for the low-level motion control of the manipulator (see also Fig. 2, where a closed-loop inverse kinematics, detailed in next subsections, is sketched). With this approach, however, the problem of handling kinematic singularities is not addressed and their avoidance cannot be guaranteed.

B. Augmented Jacobian

Another approach to redundancy resolution is the augmented Jacobian [11]. In this case, a constraint task is added to the end-effector task so as to obtain a square Jacobian matrix which can be inverted.

The main drawback of this technique is that new singularities may arise in configurations in which the end-effector Jacobian $\boldsymbol{J}$ is still full rank. Those singularities, named *algorithmic singularities*, occur when the additional task does cause conflict with the end-effector task.

A similar approach, with the same drawback, is the *extended Jacobian* approach.

C. Task Priority Redundancy Resolution

Solving the (8) in terms of a minimization problem of the quadratic cost function $\boldsymbol{\zeta}^T\boldsymbol{\zeta}$ gives the general solution [13]

$$\boldsymbol{\zeta}_r = \boldsymbol{J}^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\dot{\boldsymbol{x}}_{E,d} + \left(\boldsymbol{I}_N - \boldsymbol{J}^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\boldsymbol{J}(\boldsymbol{\eta}, \boldsymbol{q})\right)\boldsymbol{\zeta}_a \tag{11}$$

where $N = 6+n$ and $\boldsymbol{\zeta}_a \in \mathbb{R}^{6+n}$ is an arbitrary vehicle/joint velocity vector.

It can be recognized that the operator $\left(\boldsymbol{I}_N - \boldsymbol{J}^{\dagger}(\boldsymbol{\eta}, \boldsymbol{q})\boldsymbol{J}(\boldsymbol{\eta}, \boldsymbol{q})\right)$ projects a generic joint velocity vector in the null space of the Jacobian matrix. This corresponds to generate an internal motion of the manipulator arm that does not affect the end-effector motion.

Solution (11) can be seen in terms of projection of a secondary task, described by $\boldsymbol{\zeta}_a$, in the null space of the higher priority primary task, i.e., the end-effector task. A first possibility is to choose the vector $\boldsymbol{\zeta}_a$ as the gradient of a scalar objective function $H(\boldsymbol{q})$ in order to achieve a local minimum [13]

$$\boldsymbol{\zeta}_a = -k_H \nabla H(\boldsymbol{q}) \tag{12}$$

where k_H is a scalar gain factor. Another possibility is to chose a primary task $\boldsymbol{x}_{p,d} \in \mathbb{R}^m$ and a correspondent Jacobian matrix $\boldsymbol{J}_p(\boldsymbol{q}) \in \mathbb{R}^{m\times(6+n)}$

$$\dot{\boldsymbol{x}}_{p,d} = \boldsymbol{J}_p(\boldsymbol{q})\boldsymbol{\zeta}. \tag{13}$$

and to design a secondary task $\boldsymbol{x}_{s,d} \in \mathbb{R}^r$ and a correspondent Jacobian matrix $\boldsymbol{J}_s(\boldsymbol{q}) \in \mathbb{R}^{r\times(6+n)}$

$$\dot{\boldsymbol{x}}_{s,d} = \boldsymbol{J}_s(\boldsymbol{q})\boldsymbol{\zeta}. \tag{14}$$

for which the vector of joint velocity is then given by [14], [17]

$$\boldsymbol{\zeta}_r = \boldsymbol{J}_p^{\dagger}\dot{\boldsymbol{x}}_{p,d} + \left(\boldsymbol{J}_s\left(\boldsymbol{I}_N - \boldsymbol{J}_p^{\dagger}\boldsymbol{J}_p\right)\right)^{\dagger}\left(\dot{\boldsymbol{x}}_{s,d} - \boldsymbol{J}_s\boldsymbol{J}_p^{\dagger}\dot{\boldsymbol{x}}_{p,d}\right). \tag{15}$$

However, for this solution too, still remains the problem of the algorithmic singularities. In this case, it is possible to experience an algorithmic singularity when $\boldsymbol{J}_s$ and $\boldsymbol{J}_p$ are full rank but the matrix $\boldsymbol{J}_s\left(\boldsymbol{I}_N - \boldsymbol{J}_p^{\dagger}\boldsymbol{J}_p\right)$ looses rank. Extension of the approach to several tasks for highly redundant systems can be achieved by generalization of (15), as described in [22].

D. Singularity-Robust Task Priority Redundancy Resolution

A robust solution to the occurrence of the algorithmic singularities is based on the following mapping [8]:

$$\boldsymbol{\zeta}_r = \boldsymbol{J}_p^{\dagger}(\boldsymbol{\eta},\boldsymbol{q})\dot{\boldsymbol{x}}_{p,d} + \left(\boldsymbol{I}_N - \boldsymbol{J}_p^{\dagger}(\boldsymbol{\eta},\boldsymbol{q})\boldsymbol{J}_p(\boldsymbol{\eta},\boldsymbol{q})\right)\boldsymbol{J}_s^{\dagger}(\boldsymbol{\eta},\boldsymbol{q})\dot{\boldsymbol{x}}_{s,d}. \tag{16}$$

This algorithm has a clear geometrical interpretation: the two tasks are separately inverted by the use of the pseudoinverse of the corresponding Jacobian; the joint velocities associated with the secondary task are further projected in the null space of the primary task $\boldsymbol{J}_p$. Similarly to [22], extension to several tasks for highly redundant systems can be easily achieved by recursive application of (16).

E. Damped Least-Squares Inverse Kinematics Algorithms

The problem of inverting ill-conditioned matrices that may occur with all the of the aforementioned algorithms can be avoided by resorting to the damped least-square inverse given by [16]

$$\boldsymbol{J}^{\#}(\boldsymbol{\eta},\boldsymbol{q}) = \boldsymbol{J}^T(\boldsymbol{\eta},\boldsymbol{q})\left(\boldsymbol{J}(\boldsymbol{\eta},\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{\eta},\boldsymbol{q}) + \lambda^2\boldsymbol{I}_m\right)^{-1} \tag{17}$$

where $\lambda \in \mathbb{R}$ is a damping factor.

In this case, the introduction of a damping factor allows to solve the problem from the numerical point of view but, on the other hand, introduces a reconstruction error in all the velocity components. Better solutions can be found with variable damping factors or damped least-squares with numerical filtering [15], [16].

F. Closed-Loop Inverse Kinematic Algorithms

The numerical implementation of the aforementioned algorithms would lead to a numerical drift when obtaining vehicle/joint positions by integrating the vehicle/joint velocities. A closed loop version of the above equations can then be adopted. Considering as primary task the end-effector position/orientation, (16), as an example, would become

$$\boldsymbol{\zeta}_r = \boldsymbol{J}^{\dagger}(\boldsymbol{\eta},\boldsymbol{q})\left(\dot{\boldsymbol{x}}_{E,d} + \boldsymbol{K}_E\boldsymbol{e}_E\right) + \left(\boldsymbol{I}_N - \boldsymbol{J}^{\dagger}(\boldsymbol{\eta},\boldsymbol{q})\boldsymbol{J}(\boldsymbol{\eta},\boldsymbol{q})\right)\boldsymbol{J}_s^{\dagger}(\boldsymbol{\eta},\boldsymbol{q})\left(\dot{\boldsymbol{x}}_{s,d} + \boldsymbol{K}_s\boldsymbol{e}_s\right) \tag{18}$$

where $\boldsymbol{e}_E$ and $\boldsymbol{e}_s$ are the numerical reconstruction errors and $\boldsymbol{K}_E \in \mathbb{R}^{m\times m}$ and $\boldsymbol{K}_s \in \mathbb{R}^{r\times r}$ are design matrix gains to be chosen so as to ensure convergence to zero of the corresponding errors.

If the task considered is position control, its reconstruction error is simply given by the difference between the desired and the reconstructed values. In case of the orientation, however, care in the definition of such error is required to ensure convergence to the desired value. In this paper, the quaternion attitude representation is used [18]; the vector $\boldsymbol{e}_E$ for the task defined in (8) is then given by [5], [9]

$$\boldsymbol{e}_E = \begin{bmatrix} \boldsymbol{\eta}_{ee1,d} - \boldsymbol{\eta}_{ee1,r} \\ \eta_r\boldsymbol{\varepsilon}_d - \eta_d\boldsymbol{\varepsilon}_r - \boldsymbol{S}(\boldsymbol{\varepsilon}_d)\boldsymbol{\varepsilon}_r \end{bmatrix} \tag{19}$$

where $\mathcal{Q}_d = \{\eta_d, \boldsymbol{\varepsilon}_d\}$ and $\mathcal{Q}_r = \{\eta_r, \boldsymbol{\varepsilon}_r\}$ are the desired and reference attitudes expressed by quaternions, respectively, and $\boldsymbol{S}(\cdot)$ is the matrix operator performing the cross product.

In Fig. 2, the block scheme of the closed-loop inverse kinematic (CLIK) algorithm within the overall system is shown. It can be recognized that two loops are closed: a kinematic loop, around the inverse kinematics scheme and a dynamic loop that receives, as reference inputs, the joint reference trajectory output by the inverse kinematics algorithm. For the seek of simplicity, in the figure, only the primary task is shown.

Remarkably, all those inverse kinematics approaches are suitable for real-time implementation. Of course, depending on the specific algorithm, a different computational load is required [8].

IV. Application of Kinematic Control to UVMSs

State-of-art underwater manipulation missions mainly resort to manipulators mounted on remotely operated vehicles (ROVs). In this case, while the vehicle is kept in hovering, an operator drives a master–slave manipulator system in order to achieve the desired task. In view of the realization of autonomous missions of UVMSs, a kinematic control has to be developed so as to exploit the high redundancy provided by such systems and to take into account the possible redundancy of the manipulator itself as in, e.g., SAUVIM [25] and Vortex/PA10 [6].

To achieve an effective coordinated motion of the vehicle and manipulator while exploiting the redundant DOFs available, we resort to the singularity-robust task-priority redundancy resolution technique. The velocity vector $\boldsymbol{\zeta}_r$ is then computed as shown in (18).

The obtained $\boldsymbol{\zeta}_r$ can then be used to compute the position and orientation of the vehicle $\boldsymbol{\eta}_r$ and the manipulator configuration $\boldsymbol{q}_r$

$$\begin{bmatrix} \boldsymbol{\eta}_r(t) \\ \boldsymbol{q}_r(t) \end{bmatrix} = \int_0^t \begin{bmatrix} \dot{\boldsymbol{\eta}}_r(\sigma) \\ \dot{\boldsymbol{q}}_r(\sigma) \end{bmatrix} d\sigma + \begin{bmatrix} \boldsymbol{\eta}(0) \\ \boldsymbol{q}(0) \end{bmatrix} = \int_0^t \boldsymbol{J}_k^{-1}(\sigma)\boldsymbol{\zeta}_r(\sigma)d\sigma + \begin{bmatrix} \boldsymbol{\eta}(0) \\ \boldsymbol{q}(0) \end{bmatrix}. \tag{20}$$

As is customary in kinematic control approaches, the output of the above inverse kinematics algorithm provides the reference values to the dynamic control law of the vehicle-manipulator system (see Fig. 2). This dynamic control law will be in charge of computing the driving forces, i.e., the vehicle thrusters and the manipulator torques. The kinematic control algorithm is independent from the dynamic control law as long as the latter is a vehicle/joint space-based control, i.e., it requires as input the reference vehicle-joint position and velocity. In the literature number of such control laws have been proposed, e.g., [1], that are suitable to be used within the proposed kinematic control approach.

In the case of a UVMS, the primary task vector will usually include the end-effector task vector, while the secondary task vector might include the vehicle position coordinates. This choice is aimed at achieving station keeping of the vehicle as long as the end-effector task can be fulfilled with the sole manipulator arm. It is worth noticing that this approach is conceptually similar to the macro–micro manipulator approach [11]; the main difference is that the latter requires dynamic compensation of the whole system while the former is based on a kinematic control approach. This is advantageous for underwater applications in which uncertainty on dynamic parameters is experienced.

V. Fuzzy Inverse Kinematics

Because of the different inertia characteristics of the vehicle and of the manipulator, it would be preferable to perform fast motions of small amplitude by means of the manipulator while leaving the vehicle the execution of slow gross motions. This might be achieved by adopting a weighted pseudoinverse $\boldsymbol{J}_{\boldsymbol{W}}^{\dagger}$

$$\boldsymbol{J}_{\boldsymbol{W}}^{\dagger} = \boldsymbol{W}^{-1}\boldsymbol{J}^{T}\left(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^{T}\right)^{-1} \tag{21}$$

with the $(6+n)\times(6+n)$ matrix $\boldsymbol{W}^{-1}$

$$\boldsymbol{W}^{-1}(\beta) = \begin{bmatrix} (1-\beta)\boldsymbol{I}_6 & \boldsymbol{O}_{6\times n} \\ \boldsymbol{O}_{n\times 6} & \beta\boldsymbol{I}_n \end{bmatrix} \tag{22}$$

where β is a weight factor belonging to the interval [0, 1] such that $\beta = 0$ corresponds to sole vehicle motion and $\beta = 1$ to sole manipulator motion.

During the task execution, setting a constant value of β would mean to fix the motion distribution between the vehicle and the manipulator. Nevertheless, the use of a fixed weight factor inside the interval [0, 1] has a drawback: it causes motion of the manipulator also if the desired end-effector posture is out of reach; on the other hand, it causes motion of the vehicle also if the manipulator alone could perform the task.

Another problem is the necessity to handle a large number of variables; UVMSs, in fact, are complex systems and several variables must be monitored during the motion, e.g., the manipulator manipulability, the joint range limits to avoid mechanical breaks, the vehicle roll and pitch angles for correct tuning of the proximity sensors, the yaw angle to exploit the vehicle shape in presence of ocean current, etc. As can be easily understood, it is quite difficult to handle all these terms without a kinematic control approach. Nevertheless, the existing techniques do not allow to find a flexible and reliable solution.

To overcome this drawback a fuzzy theory approach has then been considered at two different levels. First, we want to manage the distribution of motion between the vehicle and the manipulator; second, we want to consider multiple secondary tasks that are activated only when the corresponding variable is outside (inside) a desired range. This can be done using different weight factors adjusted online according to the Mamdani fuzzy inference system [10], shown in Fig. 3.

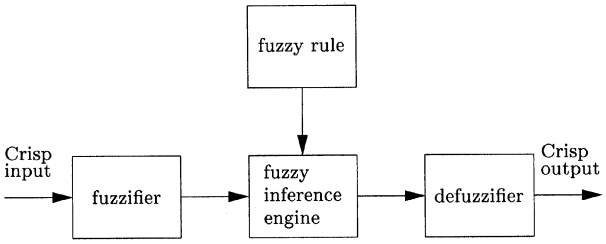

Fig. 3. Mamdani fuzzy inference system.

In detail, the crisp outputs are the scalar β of (22) that distributes the desired end-effector motion between the vehicle and the manipulator and a vector of coefficients α_i that are used in the task priority equation as follows:

$$\boldsymbol{\zeta} = \boldsymbol{J}_{\boldsymbol{W}}^{\dagger}\left(\dot{\boldsymbol{x}}_{E,d} + \boldsymbol{K}_E\boldsymbol{e}_E\right) + \left(\boldsymbol{I} - \boldsymbol{J}_{\boldsymbol{W}}^{\dagger}\boldsymbol{J}_{\boldsymbol{W}}\right)\left(\sum_i \alpha_i \boldsymbol{J}_{s,i}^{\dagger}\boldsymbol{w}_{s,i}\right) \tag{23}$$

where $\boldsymbol{w}_{s,i}$ are suitably defined secondary task variables and $\boldsymbol{J}_{s,i}$ are the corresponding Jacobians. Both β and α_i's are tuned according to the state of the system and to given behavioral rules. The inputs of the fuzzy inference system depend on the variables of interest in the specific mission. As an example, the end-effector error, the ocean current measure, the system's dexterity, the force sensor readings, can be easily taken into account by setting up a suitable set of fuzzy rules.

To avoid the exponential growth of the fuzzy rules to be implemented as the number of tasks is increased, the secondary tasks are suitably organized in a hierarchy. Also, the rules have to guarantee that only one α_i is high at a time to avoid conflict between the secondary tasks. An example of application of the approach is described in the case study.

VI. Simulation Results

The proposed technique has been verified in a number of case studies. An UVMS has been considered constituted by a vehicle with the size of the NPS AUV II [12] and a manipulator mounted on the bottom of the vehicle. The kinematics of the manipulator considered is that of the SMART-3S manufactured by COMAU. Its Denavit-Hartenberg parameters are given in Table I. The overall system, thus, has 12 DOFs. Fig. 4 shows the configuration in which all the joint positions are zero according to the used convention.

The simulations are aimed at proving the effectiveness of the fuzzy kinematic control approach; for seek of clarity, thus, only the kinematic loop performance is shown in this paper (see Fig. 2). The real vehicle/joint position will be affected by a larger error since the tracking error too has to be considered. It is worth noticing that, as long as the law level dynamic controller is suitably designed, this tracking error is bounded. Moreover, it does not affect the kinematic loop performance.

TABLE I
D-H PARAMETERS [M,RAD] OF THE MANIPULATOR MOUNTED ON THE UNDERWATER VEHICLE

	a	d	θ	α
joint 1	0.150	0	q_1	$-\pi/2$
joint 2	0.610	0	q_2	0
joint 3	0.110	0	q_3	$-\pi/2$
joint 4	0	0.610	q_4	$\pi/2$
joint 5	0	-0.113	q_5	$-\pi/2$
joint 6	0	0.103	q_6	0

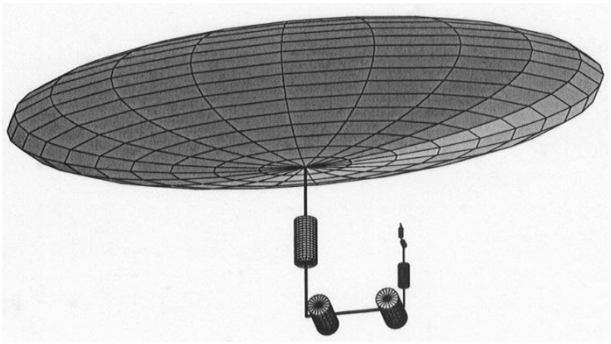

Fig. 4. UVMS in the configuration with null joint positions.

The primary task is to track a position/orientation trajectory of the end effector. The system starts from the initial configuration

$$\boldsymbol{\eta} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \qquad \text{m,deg}$$
$$\boldsymbol{q} = [0 \quad -30 \quad -110 \quad 0 \quad -40 \quad 90]^T \qquad \text{deg}$$

that corresponds to the end-effector position $\boldsymbol{\eta}_{ee1} = [0.986 \quad -0.113 \quad 2.996]^T$ m and orientation $\boldsymbol{\eta}_{ee2} = [0 \quad 0 \quad -90]^T$ deg. The end effector has to track a segment of -30 cm along z, stop there, and track a segment of 1 m along x. Both segments have to be executed with a quintic polynomial time law in 12 s. During the translation, the end effector orientation has to be kept constant. The initial configuration and the desired path are shown in Fig. 5. The duration of the simulation is 50 s. The algorithm is implemented at a sampling frequency of 20 Hz. Notice that the desired path cannot be tracked by the manipulator alone since it goes outside of its workspace. It is then necessary, somehow, to move the vehicle as well. Finally, the task is to be executed in real-time; no offline knowledge of the task is available.

Different simulations will be shown.

Case Study n. 1) Simple pseudoinversion of (8);

Case Study n. 2) Pseudoinversion of (8) by the use of a wheighted pseudinverse;

Case Study n. 3) Singolarity robust task priority algorithm;

Case Study n. 4) Integration of the former algorithm with the proposed fuzzy technique in presence of several secondary tasks.

For all the simulations a CLIK algorithm is considered, moreover, the orientation error is represented by the use of quaternions. The desired orientation, however, is still assigned in terms of Euler angles, since the transformation from Euler angles to quaternions is free from representation singularities.

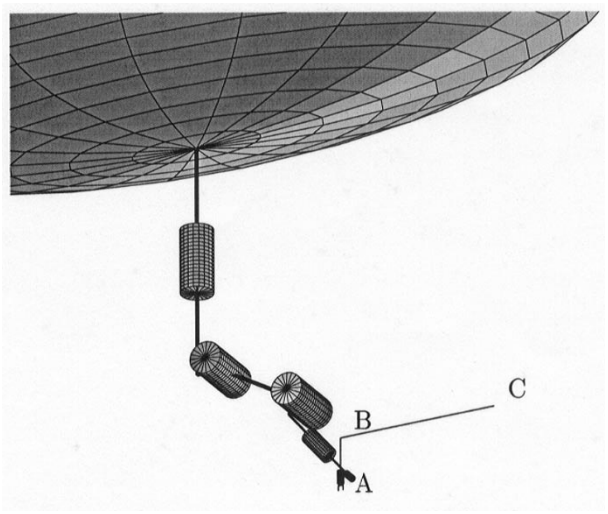

Fig. 5. Initial configuration of the UVMS. Desired end-effector position: at the start time (A); after the first movement of 12 s (B); at the final time (C).

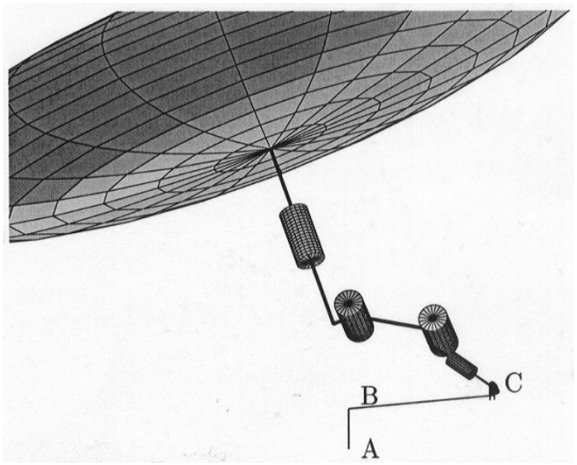

Fig. 6. Final configuration of the UVMS for the first case study. The redundancy is not exploited and the possible occurrence of undesired configurations is not avoided.

A. Case Study n. 1

The first simulation has been run by the use of a simple pseudoinversion of (8) with CLIK gain

$$\boldsymbol{K}_E = \text{blockdiag}\{1.6\boldsymbol{I}_{3\times 3} \quad 3\boldsymbol{I}_{3\times 3}\}.$$

This solution is not satisfactory in presence of such a large number of degrees of freedom. The system, in fact, is not taking into account the different nature of the degrees of freedom (vehicle and manipulator) leading to evident drawbacks: large movement of the vehicle in position and orientation, final configuration not suitable for sensor tuning, possible occurrence of kinematic singularities or joint mechanical limits. As an example, Fig. 6 reports the sketch of the final configuration where the bottom sonar would not work properly since the pitch is $\approx$ 18 deg.

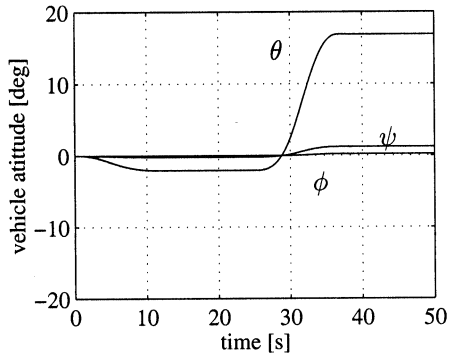

Fig. 7. Case study n. 2. Vehicle attitude in terms of Euler angles. Despite the weight factor, the vehicle can reach nondexterous configurations.

B. Case Study n. 2

In the second case study a wheighted pseudinverse is added in order to redistribute the motion between vehicle and manipulator including a cost factor that can be considered, e.g., proportional to the ratio of their inertias. The following matrix of gain has been used:

$$\boldsymbol{W}^{-1} = \text{blockdiag}\{0.01\boldsymbol{I}_{6\times 6} \quad 0.99\boldsymbol{I}_{6\times 6}\}.$$

Despite the much different costs of the two movements, the vehicle is still required to move in order to contribute to the end effector motion. However, it would be preferable to move the vehicle only when strictly necessary leading to sole movement of the manipulator in ordinary working conditions. As an example, the vehicle attitude, in terms of Euler angles, for the last simulation is shown in Fig. 7. The vehicle still has a pitch of about 20°.

C. Case Study n. 3

The drawback shown by the algorithm as presented in the Case Study n. 2 can be easily avoided by resorting to a singularity-free task-priority redundancy resolution [8]. The same task is now simulated with the introduction of the secondary task

$$\boldsymbol{x}_s = \begin{bmatrix} \phi \\ \theta \end{bmatrix}$$

with $\boldsymbol{x}_{s,d} = [0 \quad 0]^T$, meaning that the vehicle has to maintain an horizontal configuration all along the task execution. Its Jacobian is simply given by

$$\boldsymbol{J}_s = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Notice that, for this simple matrix, it is $\boldsymbol{J}_s^{\sharp} = \boldsymbol{J}_s^T$. Obviously, the vehicle position and yaw are not limited.

Fig. 8 reports the sketch of the final configuration; it can be observed that the pitch is now close to zero, as can be seen also from Fig. 9.

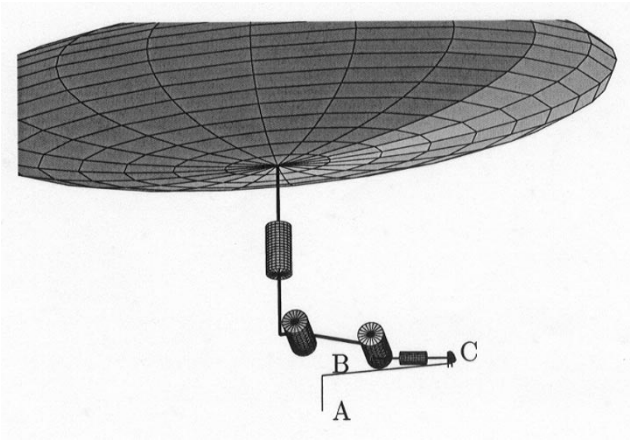

Fig. 8. Case study n. 3. Final configuration; the roll and pitch are now kept close to zero by exploiting the redundancy with the singularity-robust task priority algorithm.

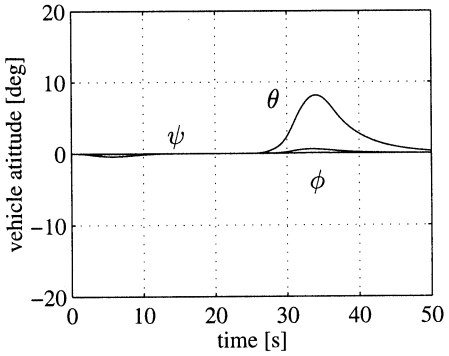

Fig. 9. Case study n. 3. Vehicle attitude in terms of Euler angles.

D. Case Study n. 4

In this simulation, the implementation of the proposed kinematic control approach is presented. In an UVMS, several variables are of interest to achieve a successful mission

- avoidance of kinematic singularities;
- keeping the joints far from the mechanical limits;
- keeping the vehicle with small roll and pitch;
- avoidance of obstacles;
- alignment of the vehicle fore–aft direction with the ocean current.

It can be observed that some of the above items are critical; the alignment with the current, however, can be significant in order to reduce power consumption [4]. As an example, in the previous case study, the use of a weight factor requires now a larger movement of the manipulator. This can lead to the occurrence of kinematic singularities. In fact, if the trajectory is assigned in real time, it is possible that the manipulator is asked to move to the border of its workspace, where the possibility to experience a kinematic singularity is high. Also, when the manipulator is outstretched mechanical joint limits can be encountered. In the simulations, the following joint limits have been assumed:

$$\boldsymbol{q}_{\min} = [-100 \quad -210 \quad -210 \quad -150 \quad -80 \quad -170]^T \text{ deg}$$
$$\boldsymbol{q}_{\max} = [100 \quad 30 \quad 10 \quad 150 \quad 80 \quad 170]^T \text{ deg}.$$

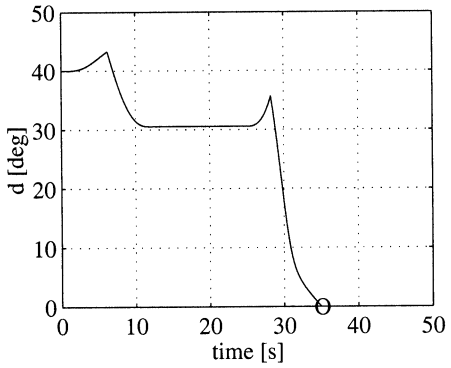

Fig. 10. Case study n. 3. Minimum distance of the 6 joints from their mechanical limits. It can be observed that large movements of the manipulator may cause hitting of the joint limits.

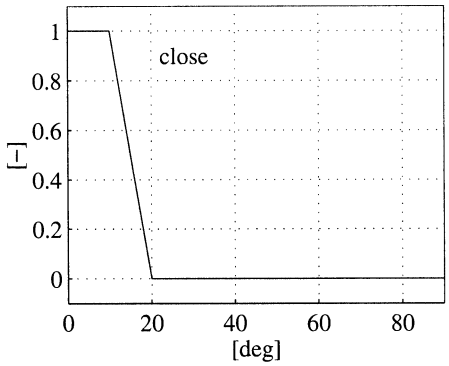

Fig. 11. Membership function of the linguistic variable: *joint limits*.

In Fig. 10, the minimum distance to such limits for the previous case study is shown. It can be observed that the manipulator hits a mechanical limit at $t \approx 35$ s.

In order to match all the constraints of such problem, a solution as been proposed based on the use of the singularity-robust task-priority merged with fuzzy techniques.

Let us consider the following tasks.

- **End-effector position/orientation:** The primary task is given, as for the previous simulations by the end-effector position and orientation. The corresponding Jacobian $\boldsymbol{J}_p$ is $\boldsymbol{J}$ given in (8).
- **Manipulability:** Since the fuzzy approach tries to move the manipulator alone, its manipulability has to be checked. A computationally limited measure of the manipulability can be obtained checking the minimum singular value of the Jacobian [14], [7]. Since the manipulability function is strongly nonlinear it is possible to adopt the following approach: when close to a singular configuration, the system tries to reconfigure itself in a dexterous configuration. The task, thus, is a nominal manipulator configuration whose Jacobian is given by

$$\boldsymbol{J}_{s1} = [\boldsymbol{O}_{6\times 6} \quad \boldsymbol{I}_6].$$

- **Mechanical limits:** Due to the mechanical structure, each joint has a limited allowed range. In case of a real-time trajectory, avoidance of such limits is crucial. For this reason the minimum distance from a mechanical limit is considered as another secondary task. Notice that the Jacobian is equal to the provious task

$$\boldsymbol{J}_{s2} = [\boldsymbol{O}_{6\times 6} \quad \boldsymbol{I}_6].$$

- **Vehicle attitude:** As for the previous cases, the vehicle attitude (roll and pitch angles) has to be kept null when possible. The Jacobian is then given by

$$\boldsymbol{J}_{s3} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Due to the simple structure of the matrices, the pseudoinversion of the secondary tasks is trivial: $\boldsymbol{J}_{s1}^{\dagger} = \boldsymbol{J}_{s1}^{T}$, $\boldsymbol{J}_{s2}^{\dagger} = \boldsymbol{J}_{s2}^{T}$, $\boldsymbol{J}_{s3}^{\dagger} = \boldsymbol{J}_{s3}^{T}$.

As shown in Section V, the aforementioned tasks are activated by fuzzy variables. The fuzzy inference system has three inputs, namely, a measure of the robot manipulability, a measure of the distance from the joints limits and a measure of the vehicle attitude. Hence, we have three linguistic variables that can take the values: *manipulability* = {singular, not singular}, *joint limits* = {close, not close} and *vehicle attitude* = {small, not small}. The output is given by the linguistic variables β and the three α_i's. The latter can take the following values: α_i = {high, low}. The linguistic variable β, named *motion* can take the following values: *motion* = {vehicle, manipulator}. As an example, the membership function of the linguistic variable *joint limits* is reported in Fig. 11.

The FIS output are considered at two different levels. The variable *motion* [β in (22)] can be considered at a higher level with respect to the variables α_i's. Roughly speaking, the motion has to be normally assigned to the manipulator, i.e., if all the α_i's are null. Hence, the value of β is related to the value of the α_i's and the first set of rules to be designed concerns the three linguistic variables α_1, α_2, and α_3. A complete and consistent set of fuzzy rules for three linguistic variables each of them defined in two fuzzy sets requires 2^3 rules for every output leading to 32 total rules. The rules have been arranged in a hierarchical structure that gives higher priority to the kinematic singularity of the manipulator and lower to the vehicle attitude. Hence, the following eight rules have been used:

1) if (manipulator is singular), then (α_1 is high);
2) if (manipulator is not singular), then (α_1 is low);
3) if (manipulator is singular), then (α_2 is low);
4) if (joint limits is not close), then (α_2 is low);
5) if (manipulator is not singular) and (joint limits is close), then (α_2 is high);
6) if (manipulator is singular) or (joint limits is close), then (α_3 is low);
7) if (vehicle attitude is small), then (α_3 is low);
8) if (manipulator is not singular) and (joint limits is not close) and (vehicle attitude is not small), then (α_3 is high).

In detail, the rules are developed as follows.

- The first two rules concern the primary task. In this case, the manipulator singularity is of concern and the variable α_1 is activated when the manipulator is close to a singularity.

TABLE II
EXAMPLES OF THE FUZZY SET RULES FOR TWO TASKS: u_1 IS THE INPUT OF A GENERIC TASK OF HIGHER PRIORITY WITH RESPECT TO u_2 CORRESPONDING TO A SECONDARY TASK. α_1 AND α_2 ARE THE CORRESPONDING OUTPUT

	$u_1 = \text{low}$	$u_1 = \text{high}$
$u_2 = \text{low}$	$\alpha_1 = \text{low}$ $\alpha_2 = \text{low}$	$\alpha_1 = \text{high}$ $\alpha_2 = \text{low}$
$u_2 = \text{high}$	$\alpha_1 = \text{low}$ $\alpha_2 = \text{high}$	$\alpha_1 = \text{high}$ $\alpha_2 = \text{low}$

- A second task (α_2), with lower priority with respect to the first, has to be added. Rule n. 3, thus, is aimed at avoid activation of this task when the primary (α_1) is high.
- Rule n. 4 and n. 5 are aimed at activating α_2. Notice that the activation of α_2 is in and with the condition that do not activate the higher priority task (α_1).
- Repeat for the third task in order of priority the same rules as done for the second by taking into account that two tasks are now of higher priority task.

Table II is aimed at clarifying the rules development with respect to two tasks, 1 and 2 in which the first is of higher priority with respect the second. The fuzzy sets are very simple, i.e., an input u_i high requires the activation of this task by imposing α_i high. It can be recognized, thus, that α_2 respect its lower priority.

It is worth noticing that the rules presented could be grouped, e.g., rules n. 1 and n. 3. The list presented, however, keep the logical structure used to develop the rules and should be clearer to the reader. Obviously, in the simulation the rules have been compacted. With this logical approach the rules are complete, consistent and continuous [10].

The and–or operations have been calculated by resorting to the *min–max* operations respectively, the implication-aggregation operations too have been calculated by resorting to the *min–max* operations respectively, the values of $\alpha_i \in [0,1]$ are obtained by defuzzification using the centroid technique and a normalization. Finally, the value of $\beta \in]0,1]$ is given by $\beta = 1 - \max_i(\alpha_i)$. Notice that the extremities of the range in which β is defined do not involve a singular configuration since, if $\beta = 1$ the manipulator alone is moving and it is not close to a kinematic singularity. On the other hand, it is preferable to have a certain degree of mobility of the manipulator avoiding $\beta = 0$; this is to guarantee that the manipulator reconfigures itself in a dexterous posture.

A simulation has been run with the proposed kinematic control leading to satisfactory results. Figs. 12–17 show some plots of interest. In detail, Fig. 12 shows the vehicle position and attitude, Fig. 13 the joint positions, Fig. 14 reports the variables considered as secondary tasks and the corresponding FIS outputs. It can be observed that, in the execution of the segment A-B, the vehicle is not asked to move since the manipulator is working in a safe posture; this can be observed from Fig. 14 where it can be noticed that the α_i's are null in the first part of the simulation. When $t \approx 30$ s joint 5 is approaching its mechanical limit and the corresponding α_2 is increasing, asking the vehicle

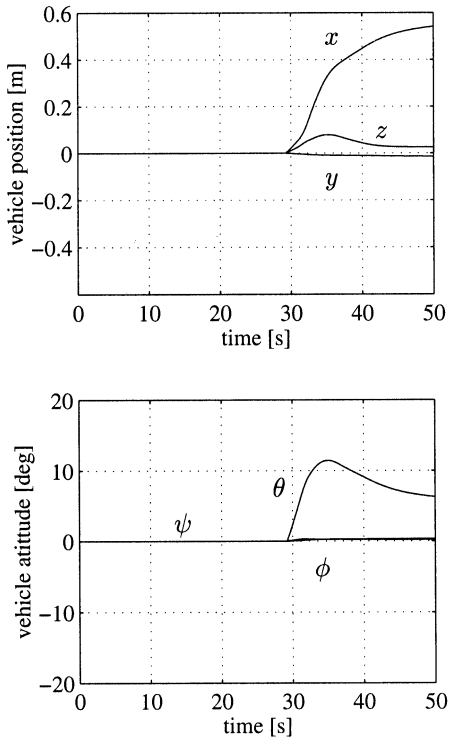

Fig. 12. Case study n. 4. Vehicle position (top) and attitude in terms of Euler angles (bottom). The movement of the vehicle is not required in the execution of the first segment (A-B, first 12 s) when the manipulator is working in dexterous configuration.

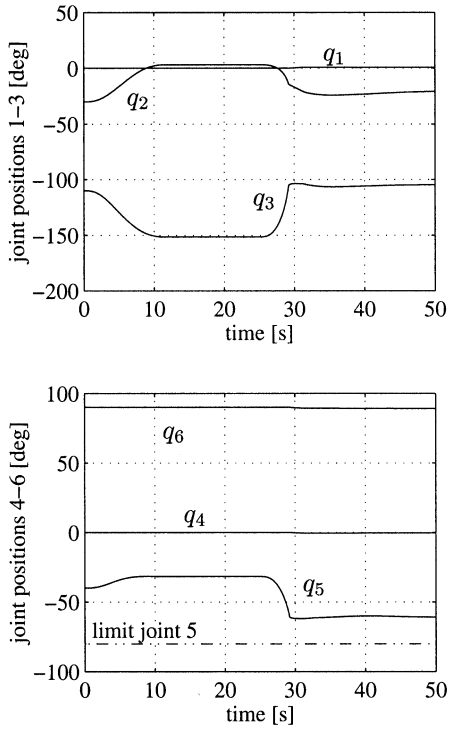

Fig. 13. Case study n. 4. Joint positions. The mechanical limit of joint 5 is highlighted; it can be observed that the system reconfigures itself in order to avoid working close to the mechanical limit.

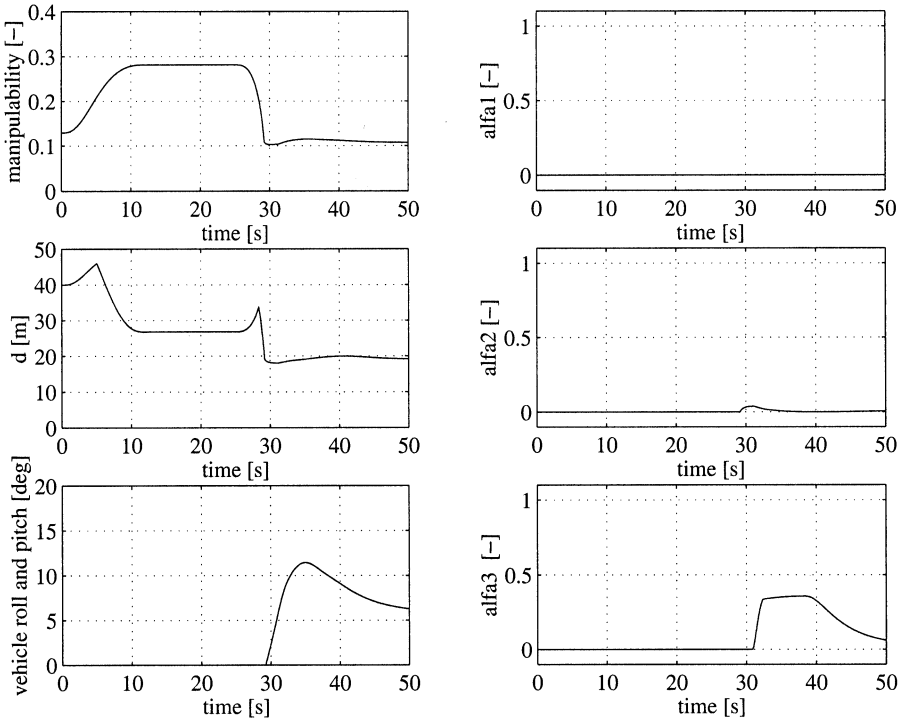

Fig. 14. Case study n. 4. Variables of interest for the secondary task (left) and output of the fuzzy inference system (right). For this specific mission, the manipulability task is not excited, the distance from the mechanical limit and the vehicle roll and pitch tasks are kept in their safe range.

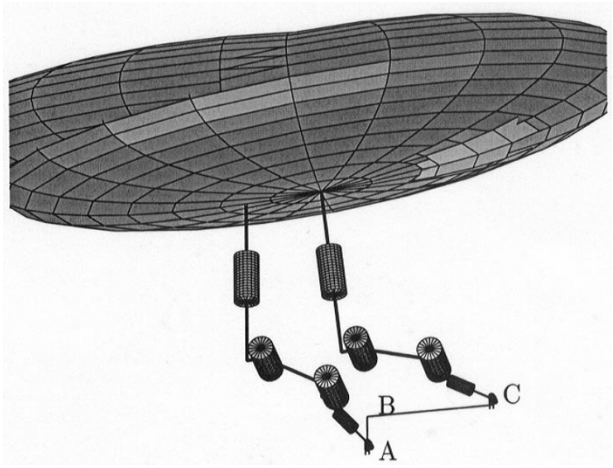

Fig. 15. Case study n. 4. Final configuration. The proposed kinematic control allows handling several variables of interest.

to contribute to the end-effector motion while the manipulator reconfigures itself; thus, always keeping a null end-effector position/orientation error, the occurrence of an hit is avoided. The same can be observed for the pitch of the vehicle; since it is at the lower hierarchical level in the FIS, its value is recovered to the null value only when the other α_i's are null.

Fig. 15 shows a sketch of the initial and final configuration of the system. Fig. 16 shows the system velocities. It can be remarked that the proposed algorithm outputs smooth trajectories. As for all the simulations shown, the end-effector position/orientation error is practically null (Fig. 17) due to the use of a CLIK algorithm.

In order to show handling of the fuzzy rules under the proposed approach while avoiding exponential growth of their number, we finally add as fourth task, specification of the vehicle yaw. Our aim is to align the vehicle fore–aft direction with the current in order to get energetic benefit from the low drag of such configuration. To limit the number of rules to be implemented we assign to this task the last priority among the secondary tasks. In this case, considering two fuzzy sets also for this last variable (*yaw* = {aligned, not aligned}), only the following three rules have to be added to the previous eight (leading to elevent rules in total instead of 64):

9) if (manipulator is singular) or (joint limits is close) or (vehicle attitude is not small), then (α_4 is low);
10) if (yaw is aligned), then (α_4 is low);
11) if (manipulator is not singular) and (joint limits is not close) and (vehicle attitude is small) and (yaw is not aligned), then (α_4 is high).

The three rules have the following aim. Rule n. 9 is aimed at giving the lower priority to this specific task. Rule n. 10 is aimed at guaranteeing that the output is always low when the corresponding input is inside the safe range. Finally, rule n. 11 activates α_4 only for the given specific combination of inputs.

VII. Conclusion

A solution to the problem of redundancy resolution and motion coordination between vehicle and manipulator in autonomous tasks of UVMSs has been addressed in this paper. UVMSs usually possess more degrees of freedom than those required to perform end-effector tasks; therefore, they are

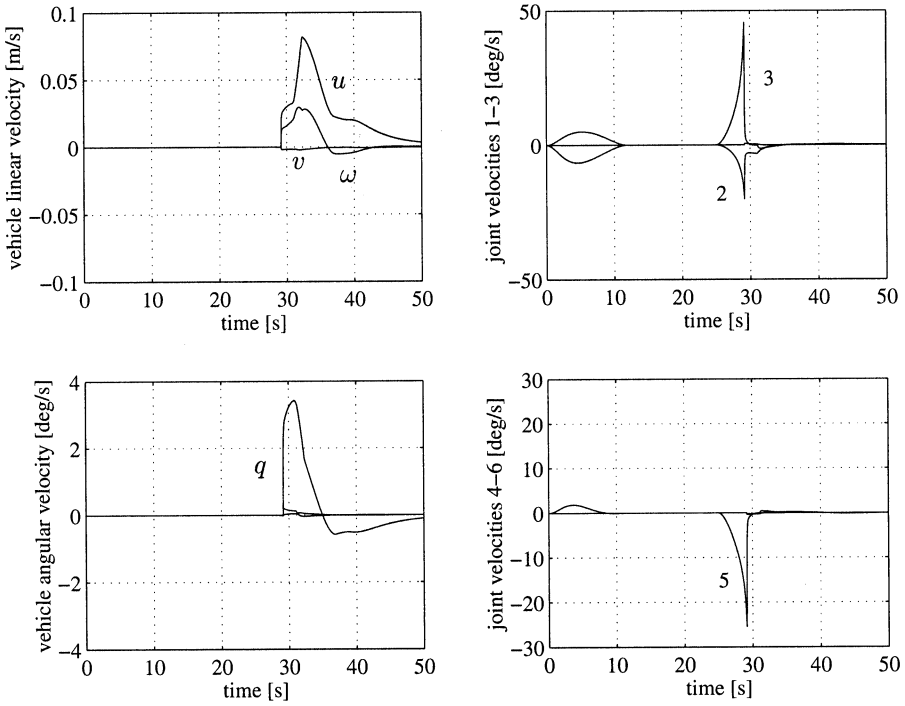

Fig. 16. Case study n. 4. System velocities.

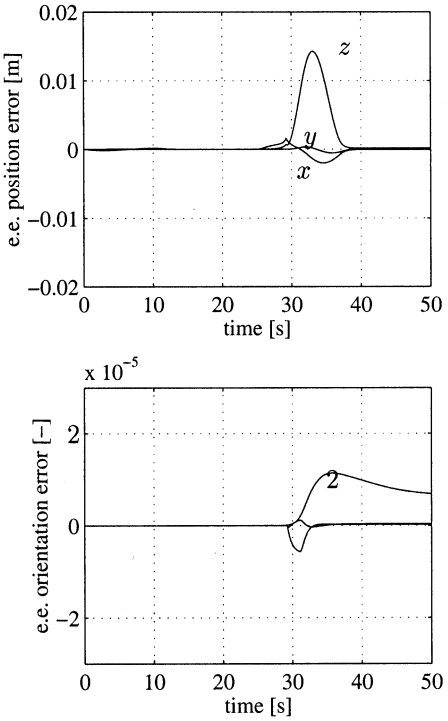

Fig. 17. Case study n. 4. End-effector position/orientation errors.

redundant systems and kinematic control techniques can be applied in order to achieve additional control objectives. In this paper, a singularity-robust task-priority inverse kinematics approach to redundancy resolution for robotic systems has been merged with a fuzzy technique to manage the vehicle-arm coordination and to handle multiple secondary tasks. Simulations with a 12-DOFs underwater vehicle-manipulator system have been developed to demonstrate effectiveness of the proposed approach. Future development might be related in implementing this algorithm in an experimental or hardware-in-the-loop vehicle-manipulator setup.

REFERENCES

[1] G. Antonelli, F. Caccavale, and S. Chiaverini, "A modular scheme for adaptive control of underwater vehicle-manipulator systems," in *Proc. 1999 Amer. Control Conf.*, San Diego, CA, June 1999, pp. 3008–3012.

[2] G. Antonelli and S. Chiaverini, "A fuzzy approach to redundancy resolution for underwater vehicle-manipulator systems," presented at the 5th IFAC Conf. Manoeuvring and Control of Marine Crafts, Aalborg, DK, Aug. 2000.

[3] ——, "Fuzzy inverse kinematics for underwater vehicle-manipulator systems," presented at the 7th Int. Symp. Advances in Robot Kinematics, Piran-Portoroz, SLO, June 2000.

[4] ——, "Task-priority redundancy resolution for underwater vehicle-manipulator systems," *Proc. 1998 IEEE Int. Conf. Robotics Automation*, pp. 768–773, May 1998.

[5] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Resolved-acceleration control of robot manipulators: A critical review with experiments," *Robotica*, vol. 16, pp. 565–573, 1998.

[6] C. Canudas de Wit, E. O. Diaz, and M. Perrier, "Robust nonlinear control of an underwater vehicle/manipulator system with composite dynamics," *Proc. 1998 IEEE Int. Conf. Robotics Automation*, pp. 452–457, May 1998.

[7] S. Chiaverini, "Estimate of the two smallest singular values of the Jacobian matrix: Application to damped least-squares inverse kinematics," *J. Robot. Syst.*, vol. 10, pp. 991–1008, 1993.

[8] ——, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 398–410, June 1997.

[9] S. Chiaverini and B. Siciliano, "The unit quaternion: A useful tool for inverse kinematics of robot manipulators," *Syst. Anal., Model., Simul.*, vol. 35, pp. 45–60, 1999.

[10] D. Driankov, H. Hellendoorn, and M. Reinfrank, *An Introduction to Fuzzy Control*. Berlin, Germany: Springer-Verlag, 1995.

[11] O. Egeland, "Task-space tracking with redundant manipulators," *IEEE Trans. Robot. Automat.*, vol. 3, pp. 471–475, June 1997.

[12] A. J. Healey and D. Lienard, "Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles," *IEEE J. Ocean. Eng.*, vol. 18, pp. 327–339, July 1993.
[13] A. Liégeois, "Automatic supervisory control of the configuration and behavior of muldibody mechanisms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 868–871, 1977.
[14] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 109–117, 1985.
[15] A. A. Maciejewski, "Numerical filtering for the operation of robotic manipulators through kinematicaly singular configurations," *J. Robot. Syst.*, vol. 5, pp. 527–552, 1988.
[16] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Trans. ASME-J. Dyna. Syst., Measure., Control*, vol. 108, pp. 163–171, 1986.
[17] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 3–15, 1987.
[18] R. E. Roberson and R. Schwertassek, *Dynamics of Multibody Systems*. Berlin, D, Germany: Springer-Verlag, 1988.
[19] N. Sarkar and T. K. Podder, "Motion coordination of underwater vehicle manipulator systems subject to drag optimization," *Proc. 1999 IEEE Int. Conf. Robotics Automation*, pp. 387–392, May 1999.
[20] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*. London, U.K.: Springer-Verlag, 2000.
[21] I. Schjølberg and T. I. Fossen, "Modeling and control of underwater vehicle-manipulator systems," in *Proc. 3rd Conf. Marine Craft Manoeuvring Control*, Southampton, U.K., 1994, pp. 45–57.
[22] B. Siciliano and J.-J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc. Int. Conf. Advanced Robotics*, Pisa, Italy, June 1991, pp. 1211–1216.
[23] T. J. Tarn and S. P. Yang, "Modeling and control for underwater robotic manipulators – An example," *Proc. 1997 IEEE Int. Conf. Robotics Automation*, pp. 2166–2171, Apr. 1997.
[24] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man Mach. Syst.*, vol. MMS-10, pp. 47–53, 1969.
[25] J. Yuh, S. K. Choi, C. Ikehara, G. H. Kim, G. McMurty, M. Ghasemi-Nejhad, N. Sarkar, and K. Sugihara, "Design of a semi-autonomous underwater vehicle for intervention missions (SAUVIM)," in *Proc. 1998 Int. Symp. Underwater Technol.*, 1998, pp. 63–68.

Gianluca Antonelli (M'02) was born in Rome, Italy, in 1970. He received the "Laurea" degree in electronic engineering and the "Research Doctorate" degree in electronic engineering and computer science, both from the University of Naples, Naples, Italy, in 1995 and 2000, respectively.

He is currently an Assistant Professor at the University of Cassino, Cassino, Italy. His research interests include simulation and control of underwater robotic systems, force/motion control of robot manipulators, path planning and obstacle avoidance for autonomous vehicles, and identification. He has published approximately 40 journal and conference papers.

Stefano Chiaverini (S'02–M'02) was born in Naples, Italy, in 1961. He received the "Laurea" and "Research Doctorate" degrees in electronics engineering from the University of Naples, Naples, Italy, in 1986 and 1990, respectively.

He is currently Professor of Automatic Control in the Engineering Faculty of the University of Cassino, Cassino, Italy, where he is Vice-Head of the Department of Automation, Electromagnetics, Information Engineering, and Industrial Mathematics. His research interests include manipulator inverse kinematics techniques, redundant manipulator control, cooperative robot systems, force/position control of manipulators, underwater robotic systems, and mobile robotic systems. He has published more than 110 international journal and conference papers and is Co-Editor of the book *Complex Robotic Systems* (New York: Springer-Verlag, 1998).

Dr. Chiaverini is currently Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION.