A Distributed Predictive Control Approach for Cooperative Manipulation of Multiple Underwater Vehicle Manipulator Systems

Shahab Heshmati-alamdari, George C. Karras and Kostas J. Kyriakopoulos

Abstract— **This paper addresses the problem of cooperative object transportation for multiple Underwater Vehicle Manipulator Systems (UVMSs) in a constrained workspace involving static obstacles. We propose a Nonlinear Model Predictive Control (NMPC) approach for a team of UVMSs in order to transport an object while avoiding significant constraints and limitations such as: kinematic and representation singularities, obstacles within the workspace, joint limits and control input saturations. More precisely, by exploiting the coupled dynamics between the robots and the object, and using certain load sharing coefficients, we design a distributed NMPC for each UVMS in order to cooperatively transport the object within the workspace's feasible region. Moreover, the control scheme adopts load sharing among the UVMSs according to their specific payload capabilities. Additionally, the feedback relies on each UVMS's locally measurements and no explicit data is exchanged online among the robots, thus reducing the required communication bandwidth. Finally, real-time simulation results conducted in UwSim dynamic simulator running in ROS environment verify the efficiency of the theoretical finding.**

I. Introduction

During the last decades, Unmanned Underwater Vehicles (UUVs) have been widely used in various applications such as marine science (e.g., biology, oceanography, archeology) and offshore industry (e.g., ship maintenance, inspection of oil/gas facilities) [1], [2]. In particular, a vast number of the aforementioned applications, demand the underwater vehicle to be enhanced with intervention capabilities as well [3], [4], thus raising increasing significant scientific interest on Underwater Vehicle Manipulator System (UVMS) lately[5]–[7]. For instance, some recent European projects: TRIDENT[8]–[11], PANDORA [12], [13], and the most recent one DexROV [14], have boosted significantly the autonomous underwater interaction tasks.

Most of the underwater manipulation tasks can be carried out more efficiently, if multiple UVMSs are cooperatively involved. On the other hand, underwater multi-robot tasks are very demanding, with the most significant challenge being imposed by the strict communication constraints [6], [15]. Therefor, employing communication based control structure in underwater environment may result in severe performance problems owing to the limited bandwidth and update rate of underwater acoustic devices. Moreover, the number of operating underwater robots in this case, is strictly limited owing to the narrow bandwidth of acoustic communication devices [16]. To overcome such limitations, recent studies on underwater cooperative manipulation are dealing with designing control schemes under lean communication requirements [17].

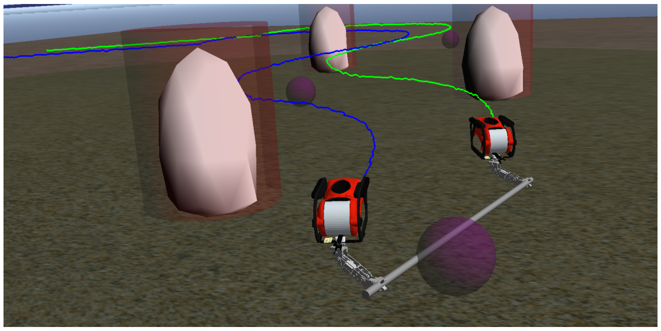

Fig. 1: Cooperatively object transportation using two UVMSs inside a constrained workspace including obstacles.

Cooperative manipulation has been well-studied in the literature, especially the centralized schemes [18]. Despite its efficiency, centralized control is less robust, and its complexity increases rapidly as the number of participating robots becomes large. On the other hand, decentralized cooperative manipulation schemes usually depend on explicit communication interchange among the robots [19]. For instance, in recent studies [20], [21], potential fields methods were employed and a multi layer control structure was developed to manage the guidance of UVMSs and the manipulation tasks. Moreover, interesting results have been given in [22]–[25] where a commonly agreed task space velocity are achieved by transferring data among the robots. However, employing the aforementioned strategies, requires each robot to communicate with the whole robot team, which consequently restricts the number of robots involved in the cooperative manipulation task owing to bandwidth limitations. Moreover, regarding cooperative manipulation, various studies can be found on the literature employing decentralized control schemes where robotic agents use only their local information or observe [26]–[28]. Most of the aforementioned studies assume that the robots are equipped with a force/torque sensor on their end effectors in order to acquire knowledge of the interaction contact forces/torques between the end effector and the common object, which may lead to a performance reduction due to sensor noise [29]–[31]. In addition, in most of the studies dealing with cooperative manipulation in literature, very important properties concerning the robotic manipulator systems such as: singular kinematic configurations of Jacobian matrix and joint limits have not been considered at all.

In this work, the problem of distributed cooperative object

The authors are with the Control Systems Lab, Department of Mechanical Engineering, National Technical University of Athens, 9 Heroon Polytechniou Street, Zografou 15780, Greece. Email: {shahab,karrasg,kkyria@mail.ntua.gr}

transportation considering multiple UVMSs in a constrained workspace with static obstacles is addressed. Specifically, given N UVMSs rigidly grasp a common object, we design distributed controllers for each UVMS in order to navigate the object from an initial position to the final one, while avoiding significant constraints and limitations such as: kinematic and representation singularities, obstacles within the workspace, joint limits and control input saturations. More precisely, by exploiting the coupled dynamics between the robots and the object and by using certain load sharing coefficients we design a distributed Nonlinear Model Predictive Control (NMPC) [32] for each UVMS in order to transport cooperatively the object and steer it along of a computed feasible path within the workspace. The design of that feasible path is based on the Navigation Function concept [33] which is adopted here in order to achieve distributed consensus on the object's desired trajectory as well to avoid collisions with the obstacles and the workspace boundary. In proposed control strategy we also take into account constraints that emanate from control input saturation as well kinematic and representation singularities. Moreover, the control scheme adopts load sharing among the UVMSs according to their specific payload capabilities. Finally, the feedback relies on each UVMS's locally measurements i.e., position and velocity measurements (e.g., sensor fusion based on measurement of various onboard sensors such as IMU, USBL and DVL) and no explicit data is exchanged online among the robots. This, consequently, increases significantly the robustness of the cooperative scheme and furthermore avoids any restrictions imposed by the acoustic communication bandwidth (e.g., the number of participating UVMSs).

II. Mathematical Modeling

Consider N UVMSs rigidly grasping an object within a constrained workspace with static obstacles (see Fig 1). We assume that each UVMS is fully-actuated at its end-effector frame. We also assume that the UVMSs are equipped with appropriate sensors, that allow them to measure their position and velocity.

A. UVMS Kinematics

Consider N UVMSs operating in a bounded workspace $\mathcal{W} \subseteq \mathbb{R}^3$. First, we denote the coordinates of each UVMS's end effector by $\boldsymbol{p}_i = [\boldsymbol{\eta}_{1,p_i}^\top, \boldsymbol{\eta}_{2,p_i}^\top]^\top$ where $\boldsymbol{\eta}_{1,p_i}^\top = [x_{p_i}, y_{p_i}, z_{p_i}]^\top$ and $\boldsymbol{\eta}_{2,p_i}^\top = [\phi_{p_i}, \theta_{p_i}, \psi_{p_i}]^\top$ denote the position and the orientation expressed in Euler angles representation with respect to (w.r.t) the inertial frame. Let $\boldsymbol{q}_i = [\boldsymbol{q}_{B,i}^\top,\ \boldsymbol{q}_{m,i}^\top]^\top \in \mathbb{R}^{n_i}$, with $n_i \in \mathbb{N}$, $i \in \mathcal{N}$ be the joint state variables of each UVMS, where $\boldsymbol{q}_{B,i} = [\boldsymbol{\eta}_{1,B_i}^\top, \boldsymbol{\eta}_{2,B_i}^\top]^\top$ is the vector that involves the position $\boldsymbol{\eta}_{1,B_i}^\top$ and the orientation $\boldsymbol{\eta}_{2,B_i}^\top$ of the vehicle and $\boldsymbol{q}_{m,i}$ is the vector of the angular positions of the manipulator's joints. Specifically, $\boldsymbol{\eta}_{1,B_i}^\top = [x_{B_i}, y_{B_i}, z_{B_i}]^\top$ and $\boldsymbol{\eta}_{2,B_i}^\top = [\phi_{B_i}, \theta_{B_i}, \psi_{B_i}]^\top,\ i \in \{O, 1, \ldots, N\}$ denote the position and the orientation expressed in Euler angles representation w.r.t the inertial frame. Let also define the UVMS' end effector generalized velocities by $\boldsymbol{v}_i = [\dot{\boldsymbol{\eta}}_{1,i}^\top, \boldsymbol{\omega}_i^\top]^\top,\ i \in \mathcal{N}$, where $\dot{\boldsymbol{\eta}}_{1,i}$ and $\boldsymbol{\omega}_i$ denote the linear and angular velocity respectively. In addition, the position and orientation of the UVMS end-effector w.r.t inertial frame, is given by the forward kinematics of the complete system (arm and vehicle base) as follows:

$$\boldsymbol{p}_i = \mathcal{F}(\boldsymbol{q}_i)\ , i \in \mathcal{N} \tag{1}$$

Moreover, for the augmented UVMS system we have [34]:

$$\boldsymbol{v}_i = \boldsymbol{J}_i(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i,\ i \in \mathcal{N} \tag{2}$$

where $\dot{\boldsymbol{q}}_i = [\dot{\boldsymbol{q}}_{B,i}^\top, \dot{\boldsymbol{q}}_{m,i}^\top]^\top \in \mathbb{R}^{n_i}$ is the velocity vector involving the velocities of the vehicle w.r.t the inertial frame as well as the joint velocities of the manipulator and $\boldsymbol{J}_i(\boldsymbol{q}_i)$ is the geometric Jacobian matrix [34]. Note that the $\boldsymbol{J}_i(\boldsymbol{q}_i)$ becomes singular at kinematic singularities defined by the set

$$Q_{s_i} = \{\boldsymbol{q}_i \in \mathbb{R}^{n_i} : \det(\boldsymbol{J}_i(\boldsymbol{q}_i)[\boldsymbol{J}_i(\boldsymbol{q}_i)]^\top) = 0\},\ i \in \mathcal{N}. \tag{3}$$

B. UVMS Dynamics

The dynamics of a UVMS after straightforward algebraic manipulations can be written as [34]:

$$\boldsymbol{M}_{q_i}(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \boldsymbol{C}_{q_i}(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{D}_{q_i}(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{g}_{q_i}(\boldsymbol{q}_i) = \boldsymbol{\tau}_i - \boldsymbol{J}_i^\top \boldsymbol{\lambda}_i \tag{4}$$

for $i \in \mathcal{N}$, where $\boldsymbol{\lambda}_i$ is the vector of generalized interaction forces and torques that UVMS exerts on the object, $\boldsymbol{\tau}_i$ denotes the vector of control inputs (forces and torques), $\boldsymbol{M}_{q_i}(\boldsymbol{q}_i)$ is the inertial matrix, $\boldsymbol{C}_{q_i}(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)$ represents coriolis and centrifugal terms, $\boldsymbol{D}_{q_i}(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)$ models dissipative effects and $\boldsymbol{g}_i(\boldsymbol{q}_i)$ encapsulates the gravity and buoyancy effects. In view of (2) we have:

$$\dot{\boldsymbol{v}}_i = \boldsymbol{J}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \dot{\boldsymbol{J}}_i(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i,\ i \in \mathcal{N} \tag{5}$$

where $\dot{\boldsymbol{J}}_i(\boldsymbol{q}_i) \in \mathbb{R}^{6 \times n_i}$ represents the Jacobian derivative function. Then, by employing the differential kinematics (2) as well as (5), we obtain from (4) the transformed task space dynamics [35]:

$$\boldsymbol{M}_i(\boldsymbol{q}_i)\dot{\boldsymbol{v}}_i + \boldsymbol{C}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\boldsymbol{v}_i + \boldsymbol{D}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\boldsymbol{v}_i + \boldsymbol{g}_i(\boldsymbol{q}_i) = \boldsymbol{u}_i - \boldsymbol{\lambda}_i \tag{6}$$

for all $i \in \mathcal{N}$ with corresponding task space terms $\boldsymbol{M}_i \in \mathbb{R}^{6\times 6}$, $\boldsymbol{C}_i \in \mathbb{R}^{6\times 6}$, $\boldsymbol{D}_i \in \mathbb{R}^{6\times 6}$, $\boldsymbol{g}_i \in \mathbb{R}^6$ with $\boldsymbol{u}_i \in \mathbb{R}^6$ to be the vector of task space generalized forces/torques. The task space dynamics (6) can be written in vector form as:

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}} + \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\boldsymbol{v} + \boldsymbol{D}(\dot{\boldsymbol{q}}, \boldsymbol{q})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u} - \boldsymbol{\lambda} \tag{7}$$

where $\boldsymbol{v} = [\boldsymbol{v}_1^\top, \ldots, \boldsymbol{v}_N^\top]^\top \in \mathbb{R}^{6N}$, $\boldsymbol{M} = \text{diag}\{[\boldsymbol{M}_i]\} \in \mathbb{R}^{6N\times 6N}$, $\boldsymbol{C} = \text{diag}\{[\boldsymbol{C}_i]\} \in \mathbb{R}^{6N\times 6N}$, $\boldsymbol{D} = \text{diag}\{[\boldsymbol{D}_i]\} \in \mathbb{R}^{6N\times 6N}$, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^\top, \ldots, \boldsymbol{\lambda}_N^\top]^\top$, $\boldsymbol{u} = [\boldsymbol{u}_1^\top, \ldots, \boldsymbol{u}_N^\top]^\top$, $\boldsymbol{d} = [\boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_N^\top]^\top$, $\boldsymbol{g} = [\boldsymbol{g}_1^\top, \ldots, \boldsymbol{g}_N^\top]^\top \in \mathbb{R}^{6N}$.

C. Object Dynamic

We denote the object's coordinate and its generalized velocities by $\boldsymbol{x}_O = [\boldsymbol{\eta}_{1,O}^\top, \boldsymbol{\eta}_{2,O}^\top]^\top$ and $\boldsymbol{v}_O = [\dot{\boldsymbol{\eta}}_{1,O}^\top, \boldsymbol{\omega}_O^\top]^\top$ respectively, with $\boldsymbol{\eta}_{1,O}^\top = [x_O, y_O, z_O]^\top$ and $\boldsymbol{\eta}_{2,O}^\top = [\phi_O, \theta_O, \psi_O]^\top$. The dynamics of the object can be given [34]:

$$\dot{\boldsymbol{x}}_O = \boldsymbol{J}_O'(\boldsymbol{\eta}_{2,O})^{-1}\boldsymbol{v}_O \tag{8a}$$

$$\boldsymbol{M}_O(\boldsymbol{x}_O)\dot{\boldsymbol{v}}_O + \boldsymbol{C}_O(\boldsymbol{v}_O, \boldsymbol{x}_O)\boldsymbol{v}_O + \boldsymbol{D}_O(\boldsymbol{v}_O, \boldsymbol{x}_O)\boldsymbol{v}_O + \boldsymbol{g}_O = \boldsymbol{\lambda}_O \tag{8b}$$

where $\boldsymbol{M}_O(\boldsymbol{x}_O)$ is the positive definite inertia matrix, $\boldsymbol{C}_O(\boldsymbol{v}_O, \boldsymbol{x}_O)$ is the Coriolis matrix, $\boldsymbol{g}_O$ is the vector of gravity and buoyancy effects, $\boldsymbol{D}_O(\boldsymbol{v}_O, \boldsymbol{x}_O)$ models dissipative effects and $\boldsymbol{\lambda}_O$ is the vector of generalized forces acting on the object's center of mass. Moreover, $\boldsymbol{J}'_O(\boldsymbol{\eta}_{2,O})$ is the object representation Jacobian that transforms the Euler angle rates into velocity $\boldsymbol{\omega}_O$ and is singular when when $\theta_O = \pm\frac{\pi}{2}$[34].

III. Control Methodology

First, the overall dynamics of the system are formulated which are decoupled next among the object and the robots by using certain load sharing coefficients. Each UVMS at each sampling time, solves a NMPC subject to its corresponding part of that overall dynamics and a number of inequality constraints that incorporate its internal limitations (e.g., joint limits, kinematic and representation singularities, collision between the arm and the base, manipulability) in order to drive cooperatively the object and steer it along of a computed feasible path within the workspace. The computation of that feasible path is based on the concept of Navigation Functions [33] that is incorporated to deal with consensus on a mutually agreed trajectory of the commonly object.

A. Coupled Dynamics

Owing to the rigid grasp of the object, the following equations hold:

$$\boldsymbol{p}_i = \boldsymbol{x}_O + \begin{bmatrix} {}^I\boldsymbol{R}_O \boldsymbol{l}_i \\ \boldsymbol{\alpha}_i \end{bmatrix}, \ i \in \mathcal{N} \tag{9}$$

where the vectors $\boldsymbol{l}_i = [l_{ix}, l_{iy}, l_{iz}]^\top$ and $\boldsymbol{\alpha}_i = [\alpha_{ix}, \alpha_{iy}, \alpha_{iz}]^\top, \ i \in \mathcal{N}$ represent the *constant* relative position and orientation of the end-effector w.r.t the object, expressed in the object's frame and ${}^I\boldsymbol{R}_O$ denotes the rotation matrix between the object and the inertial frame $\{I\}$. Thus, using (9) each UVMS can compute the object's position w.r.t inertial frame $\{I\}$, since the object geometric parameters are considered known. Furthermore, due to the grasping rigidly, it holds that $\boldsymbol{\omega}_i = \boldsymbol{\omega}_O, \ i \in \mathcal{N}$, one obtains:

$$\boldsymbol{v}_O = \boldsymbol{J}_{i_O} \boldsymbol{v}_i, \ i \in \mathcal{N} \tag{10}$$

where $\boldsymbol{J}_{i_O}, \ i \in \mathcal{N}$ denotes the Jacobian from the end-effector of each UVMS to the object's center of mass, that is defined as:

$$\boldsymbol{J}_{i_O} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & -\boldsymbol{S}(\boldsymbol{l}_i) \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \in \mathbb{R}^{6\times6}, \quad i \in \mathcal{N}$$

where $\boldsymbol{S}(\boldsymbol{l}_i)$ is the skew-symmetric matrix of vector $\boldsymbol{l}_i = [l_{ix}, l_{iy}, l_{iz}]^\top$. Notice that $\boldsymbol{J}_{i_O}, \ i \in \mathcal{N}$ are always full-rank owing to the grasp rigidity and hence obtain a well defined inverse. Thus, the object's velocity can be easily computed via (10). Moreover, from (10), one obtains the acceleration relation:

$$\dot{\boldsymbol{v}}_O = \boldsymbol{J}_{i_O} \dot{\boldsymbol{v}}_i + \dot{\boldsymbol{J}}_{i_O} \boldsymbol{v}_i, \ i \in \mathcal{N} \tag{11}$$

which will be used in the subsequent analysis. In addition, the kineto-statics duality along with the grasp rigidity suggest that the force $\boldsymbol{\lambda}_O$ acting on the object's center of mass and the generalized forces $\boldsymbol{\lambda}_i, \ i \in \mathcal{N}$, exerted by the UVMSs at the grasping points, are related through:

$$\boldsymbol{\lambda}_O = \boldsymbol{G}^\top \boldsymbol{\lambda} \tag{12}$$

where: $$\boldsymbol{G} = \Big[[\boldsymbol{J}_{O_1}]^\top, \dots, [\boldsymbol{J}_{O_N}]^\top\Big]^\top \in \mathbb{R}^{6N\times6} \tag{13}$$

is the full column-rank grasp matrix, $\boldsymbol{J}_{O_i} = [\boldsymbol{J}_{i_O}]^{-1}, \ i \in \mathcal{N}$ and $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_{\boldsymbol{1}}^\top, \dots, \boldsymbol{\lambda}_{\boldsymbol{N}}^\top]^\top$ is the vector of overall interaction forces and torques. By substituting (7) into (12) one obtains:

$$\boldsymbol{\lambda} = \boldsymbol{G}^\top \Big[\boldsymbol{u} - \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}} - \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\boldsymbol{v} - \boldsymbol{D}(\dot{\boldsymbol{q}}, \boldsymbol{q})\boldsymbol{v} - \boldsymbol{g}(\boldsymbol{q})\Big] \tag{14}$$

which, after substituting (10), (11), (8) and rearranging terms, yields the overall system coupled dynamics:

$$\widetilde{\boldsymbol{M}}(\tilde{\boldsymbol{q}}_{ov})\dot{\boldsymbol{v}}_O + \widetilde{\boldsymbol{C}}(\tilde{\boldsymbol{q}}_{ov})\boldsymbol{v}_O + \widetilde{\boldsymbol{D}}(\tilde{\boldsymbol{q}}_{ov})\boldsymbol{v}_O + \widetilde{\boldsymbol{g}}(\tilde{\boldsymbol{q}}_{ov}) = \boldsymbol{G}^\top \boldsymbol{u} \tag{15}$$

where $\tilde{\boldsymbol{q}}_{ov} = [\boldsymbol{q}^\top, \dot{\boldsymbol{q}}^\top, \boldsymbol{x}_O^\top, \boldsymbol{v}_O^\top]^\top$ and:

$$\widetilde{\boldsymbol{M}}(\tilde{\boldsymbol{q}}_{ov}) = \boldsymbol{M}_O(\boldsymbol{x}_O) + \boldsymbol{G}^\top \boldsymbol{M}(\boldsymbol{q})\boldsymbol{G}$$
$$\widetilde{\boldsymbol{C}}(\tilde{\boldsymbol{q}}_{ov}) = \boldsymbol{C}_O(\boldsymbol{v}_O, \boldsymbol{x}_O) + \boldsymbol{G}^\top \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{G}}(\dot{\boldsymbol{q}}, \boldsymbol{q}) + \boldsymbol{G}^\top \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\boldsymbol{G}$$
$$\widetilde{\boldsymbol{D}}(\tilde{\boldsymbol{q}}_{ov}) = \boldsymbol{D}_O(\boldsymbol{v}_O, \boldsymbol{x}_O) + \boldsymbol{G}^\top \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\boldsymbol{G}, \ \widetilde{\boldsymbol{g}}(\tilde{\boldsymbol{q}}_{ov}) = \boldsymbol{g}_O(\boldsymbol{x}_O) + \boldsymbol{G}^\top \boldsymbol{g}(\boldsymbol{q})$$

Now, consider the design constants $c_i, \ i \in \mathcal{N}$ satisfying:

$$c_i \in (0,1), \forall i \in \mathcal{N} \qquad \text{and} \qquad \sum_{i\in\mathcal{N}} c_i = 1, \tag{16}$$

that we introduce here in order to act as the load sharing coefficients for the team of UVMS. In view of (16), by employing (12), (2), (5), (10) and (11), and after straightforward algebraic manipulations, the overall coupled dynamics of (15) can be divided and rewritten as:

$$\sum_{i\in\mathcal{N}} \Big\{ \widetilde{\boldsymbol{M}}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \widetilde{\boldsymbol{C}}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\dot{\boldsymbol{q}}_i + \widetilde{\boldsymbol{D}}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\dot{\boldsymbol{q}}_i + \widetilde{\boldsymbol{g}}_i(\boldsymbol{q}_i) \Big\} = \sum_{i\in\mathcal{N}} \boldsymbol{J}_{O_i}^\top \boldsymbol{u}_i \tag{17}$$

where:

$$\widetilde{\boldsymbol{M}}_i(\boldsymbol{q}_i) = c_i \boldsymbol{M}_O \boldsymbol{J}_{i_O} \boldsymbol{J}_i + \boldsymbol{J}_{O_i}^\top \boldsymbol{M}_i \boldsymbol{J}_i$$
$$\widetilde{\boldsymbol{C}}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i) = c_i \Big[\boldsymbol{M}_O \boldsymbol{J}_{i_O} \dot{\boldsymbol{J}}_i + \boldsymbol{M}_O \dot{\boldsymbol{J}}_{i_O} \boldsymbol{J}_i + \boldsymbol{C}_O \boldsymbol{J}_{i_O} \boldsymbol{J}_i\Big] + \boldsymbol{J}_{O_i}^\top \Big[\boldsymbol{M}_i \dot{\boldsymbol{J}}_i + \boldsymbol{C}_i \boldsymbol{J}_i\Big]$$
$$\widetilde{\boldsymbol{D}}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i) = c_i \boldsymbol{D}_O \boldsymbol{J}_{i_O} \boldsymbol{J}_i + \boldsymbol{J}_{O_i}^\top \boldsymbol{D}_i \boldsymbol{J}_i, \quad \widetilde{\boldsymbol{g}}_i(\boldsymbol{q}_i) = c_i \boldsymbol{g}_O + \boldsymbol{J}_{O_i}^\top \boldsymbol{g}_i$$

which is the distributed version of (15), since for each UVMS, it is based only individually on its locally measurements (i.e., $\boldsymbol{q}_i$ and $\dot{\boldsymbol{q}}_i$). Now, by using the notation $\boldsymbol{x}_i = [\boldsymbol{q}_i^\top, \dot{\boldsymbol{q}}_i^\top]^\top$, the decentralized dynamics of each UVMS based on (17), can be written as compact form:

$$\dot{\boldsymbol{x}}_i = f_i(\boldsymbol{x}_i, \boldsymbol{u}_i) = \begin{bmatrix} f_{i_1}(\boldsymbol{x}_i) \\ f_{i_2}(\boldsymbol{x}_i, \boldsymbol{u}_i) \end{bmatrix}, i \in \mathcal{N} \tag{18}$$

where:

$$f_{i_1}(\boldsymbol{x}_i) = \dot{\boldsymbol{q}}_i$$
$$f_{i_2}(\boldsymbol{x}_i, \boldsymbol{u}_i) = \widetilde{\boldsymbol{M}}_i^{\#}(\boldsymbol{q}_i)\Big(\boldsymbol{J}_{O_i}^\top(\boldsymbol{q}_i)\boldsymbol{u}_i - \widetilde{\boldsymbol{C}}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\dot{\boldsymbol{q}}_i - \widetilde{\boldsymbol{D}}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\dot{\boldsymbol{q}}_i - \widetilde{\boldsymbol{g}}_i(\boldsymbol{q}_i)\Big)$$

with: $$\widetilde{\boldsymbol{M}}_i^{\#}(\boldsymbol{q}_i) = \widetilde{\boldsymbol{M}}_i(\boldsymbol{q}_i)\Big[\widetilde{\boldsymbol{M}}_i(\boldsymbol{q}_i)\widetilde{\boldsymbol{M}}_i^\top(\boldsymbol{q}_i)\Big]^{-1}$$

B. Description of the Workspace

In this work, the obstacles, the robots as well as the workspace are all modeled by spheres (i.e., we adopt the spherical world representation [33]). In this spirit, let $\mathcal{B}(\boldsymbol{x}_O, r_0)$ be a closed ball that covers the volume of the object and has radius r_0. We also define the closed balls $\mathcal{B}(\boldsymbol{p}_i, \bar{r}), i \in \mathcal{K}$, centered at the end-effector of each UVMS that cover the robot volume for all possible configurations. We also assume that the distance among the grasping points on the given object is at least $2\bar{r}$ [1]. Furthermore, we define a ball area $\mathcal{B}(\boldsymbol{x}_O, R)$ located at $\boldsymbol{x}_O$ with radius $R = \bar{r} + r_o$

[1] The distance $2\bar{r}$ denotes the minimum allowed distance at which two bounding spheres $\mathcal{B}(\boldsymbol{p}_i, \bar{r})$ and $\mathcal{B}(\boldsymbol{p}_j, \bar{r})$ $i, j \in \mathcal{K}, i \neq j$ do not collide.

that includes the whole volume of the robotic team and the object (see Fig. 2). Finally, the $\mathcal{M}$ static obstacles are defined as closed spheres described by $\pi_m = \mathcal{B}(\boldsymbol{p}_{\pi_m}, r_{\pi_m}),\ m \in \{1, \ldots, \mathcal{M}\}$, where $\boldsymbol{p}_{\pi_m} \in \mathbb{R}^3$ is the center and the r_{π_m} the radius of the obstacle π_m.

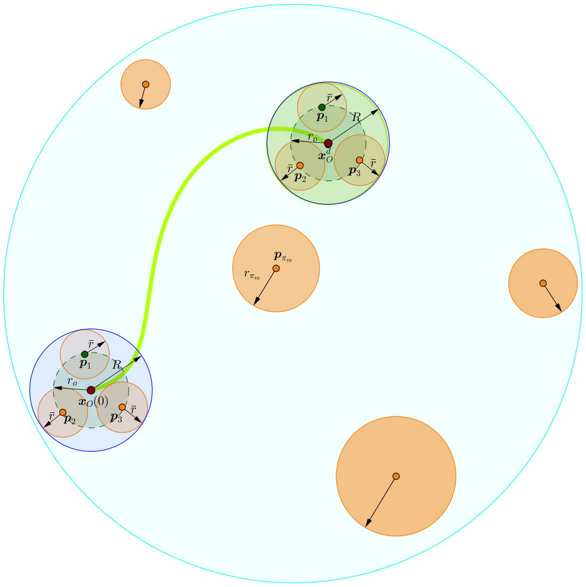

Fig. 2: Graphical representation of a safe trajectory of the robotic team. The orange areas indicate the obstacles. The blue line encircles the area covered by the robotic team and the object.

C. Navigation Function

The calculation of the desired object trajectory within the workspace $\mathcal{W}$ relies on the Navigation Function concept originally proposed by Rimon and Koditschek in [33] as follows:

$$\phi_O(\boldsymbol{x}_O; \boldsymbol{x}_O^d) = \frac{\gamma(\boldsymbol{x}_O - \boldsymbol{x}_O^d)}{[\gamma^k(\boldsymbol{x}_O - \boldsymbol{x}_O^d) + \beta(\boldsymbol{x}_O)]^{\frac{1}{k}}} \tag{19}$$

where $\phi_O : \xrightarrow{\mathcal{W} - \bigcap_{m=1}^{\mathcal{M}} \mathcal{B}(\boldsymbol{p}_{\pi_m}, r_{\pi_m})} [0, 1)$denotes the potential that derives a safe motion vector field within the free space $\mathcal{W} - \bigcap_{m=1}^{\mathcal{M}} \mathcal{B}(\boldsymbol{p}_{\pi_m}, r_{\pi_m})$. Moreover, $k > 1$ is a design constant, $\gamma(\boldsymbol{x}_O - \boldsymbol{x}_O^d) > 0$ with $\gamma(\boldsymbol{0}) = 0$ represents the attractive potential field to the goal position $\boldsymbol{x}_O^d$ and $\beta(\boldsymbol{x}_O) > 0$ [33]:

$$\lim_{\boldsymbol{x}_O \to \left\{\begin{matrix}\text{Boundary}\\ \text{Obstacles}\end{matrix}\right.} \beta(\boldsymbol{x}_O) = 0$$

represents the repulsive potential field by the workspace boundary and the obstacle regions. It was proven in [33] that $\phi_O(\boldsymbol{x}_O, \boldsymbol{x}_O^d)$ has a global minimum at $\boldsymbol{x}_O^d$ and no other local minima for sufficiently large k. Thus, a feasible path that leads from any initial obstacle-free configuration to the desired configuration might be generated by following the negated gradient of $\phi_O(\boldsymbol{x}_O, \boldsymbol{x}_O^d)$. Consequently, the object's desired motion profile is designed as follows:

$$\boldsymbol{v}_O^d(t) = -K_{NF} \boldsymbol{J}_O'(\boldsymbol{\eta}_{2,O}) \nabla_{\boldsymbol{x}_O} \phi_O(\boldsymbol{x}_O(t), \boldsymbol{x}_O^d) \tag{20}$$

where $K_{NF} > 0$ is a positive gain. Now let us define a sequence of sampling time $\{t_j\}_{j \geq 0}$ with a constant sampling time $h > 0$ with $h < T_p$ for the system such that:

$$t_{j+1} = t_j + h,\ \forall j \geq 0 \tag{21}$$

Therefore, given a current position and velocity of the object $\boldsymbol{x}_O(t_j),\ \boldsymbol{v}_O(t_j)$ at the time t_j each UVMS can propagates for time interval $s \in [t_j, t_j + T_P]$ where T_P is the prediction horizon, a map of desired trajectory and velocity of the object based on (19), (20) given as $\boldsymbol{x}_O^d(s)$ and $\boldsymbol{v}_O^d(s),\ s \in [t_j, t_j + T_P]$ which will be used in the subsequent analysis.

D. Constraints

State Constraints:

We consider for each UVMS a set of constraints which are captured by the state constraint set X_i of the system, given by:

$$\boldsymbol{x}_i(t) \in X_i \subset \mathbb{R}^{2n_i} \tag{22}$$

which is formed by the following constraints:

$$\theta_O(t) \in (-\frac{\pi}{2}, \frac{\pi}{2}) \tag{23a}$$

$$\boldsymbol{q}_i \in \mathbb{R}^{n_i} \backslash \Big(Q_{s_i}(\boldsymbol{q}_i) \cup Q_{l_i}(\boldsymbol{q}_i)\Big),\ i \in \mathcal{N} \tag{23b}$$

$$|\dot{q}_{k_i}| \leq \bar{\dot{q}}_i, \forall k \in \{1, \ldots, n_i\}, i \in \mathcal{N} \tag{23c}$$

where $Q_{s_i}(\boldsymbol{q}_i)$ is the set of singular position of the system (3) and $Q_{l_i}(\boldsymbol{q}_i)$ is the set of manipulator's joint limits given:

$$Q_{l_i}(\boldsymbol{q}_i) = \{\boldsymbol{q}_i \in \mathbb{R}^{n_i} \colon |q_{k_i}| \leq \bar{q}_{k_i}\}, \forall k \in \{1, \ldots, n\}, i \in \mathcal{N} \tag{24}$$

where $\bar{q}_{k_i}$ is the limit bound for the corresponding joint $q_{k_i},\ k \in \{1, \ldots, n\}, i \in \mathcal{N}$. Moreover, $\bar{\dot{q}}_i$ is the upper value for the joint velocity $\dot{q}_{k_i},\ k \in \{1, \ldots, n\}, i \in \mathcal{N}$. Therefore, the set X_i capture all the state constraints of the systems (18), i.e., singularity avoidance as well as joint limits limitations.

Input Constraints:

We consider the input constraints for each UVMS as:

$$||\boldsymbol{\tau}_i|| \leq \bar{\boldsymbol{\tau}}_i \Leftrightarrow ||\boldsymbol{J}_i(\boldsymbol{q}_i)^\top \boldsymbol{u}_i|| \leq \bar{\boldsymbol{\tau}}_i$$

where $\bar{\boldsymbol{\tau}}_i$ is a vector including corresponding limit bound for each actuated joint $\tau_{k_i},\ k \in \{1, \ldots, \tau_{n_i}\}, i \in \mathcal{N}$ where τ_n is the number of actuated joints. Therefore, we can define the control input set T_i:

$$\boldsymbol{\tau}_i(t) \in T_i \subset \mathbb{R}^{\tau_{n_i}} \tag{25}$$

with:

$$T_i = \{\boldsymbol{\tau}_i \in \mathbb{R}^{\tau_{n_i}} : ||\boldsymbol{J}_i(\boldsymbol{q}_i)^\top \boldsymbol{u}_i|| \leq \bar{\boldsymbol{\tau}}_i,\ \forall \boldsymbol{x}_i \in X_i\}$$

E. Control design

At each sampling time, the UVMS $i \in \mathcal{N}$ solves an NMPC scheme subject to its corresponding dynamics (18) and a number of inequality constraints (i.e., (23a)-(23c) and (25)) in order to follow the computed desired trajectory $\boldsymbol{x}_O^d(s)$ and velocity $\boldsymbol{v}_O^d(s)$ for a time interval $s \in [t_j, t_j + T_P]$ based on (19), (20) and (21). In particular, in sampled data NMPC, a Finite Horizon Optimal Control Problem (FHOCP) is solved at discrete sampling time instants t_j based on the current state measurements $\boldsymbol{x}_i(t_j),\ i \in \mathcal{N}$. For UVMS $i,\ i \in \mathcal{N}$, the open-loop input signal applied in between the sampling instants is given by the solution of the FHOCP:

$$\min_{\hat{\boldsymbol{\tau}}_i(\cdot)} J_i(\boldsymbol{x}(t_j), \hat{\boldsymbol{\tau}}_i(\cdot)) = \tag{26a}$$

$$\min_{\hat{\boldsymbol{\tau}}_i(\cdot)} \Bigg\{ \int_{t_i}^{t_i+T_p} \Big[F_i\big(\hat{\boldsymbol{x}}_O(s), \hat{\boldsymbol{v}}_O(s), \hat{\boldsymbol{\tau}}_i(s)\big)\Big] ds$$

$$+ E_i\Big(\hat{\boldsymbol{x}}_O(t_j + T_P), \hat{\boldsymbol{v}}_O(t_j + T_P)\Big)\Bigg\}$$

subject to:

$$\dot{\hat{\boldsymbol{x}}}_i(s) = f_i(\hat{\boldsymbol{x}}_i(s), \hat{\boldsymbol{u}}_i(s)), \qquad \hat{\boldsymbol{x}}_i(t_j) = \boldsymbol{x}_i(t_j), \tag{26b}$$

$$\hat{\boldsymbol{\tau}}_i(s) = \boldsymbol{J}_i^\top(\hat{\boldsymbol{q}}_i)\hat{\boldsymbol{u}}_i + \boldsymbol{\tau}_{i0}(\boldsymbol{q}_i), \;\; s \in [t_j, t_j + T_P] \tag{26c}$$

$$\hat{\boldsymbol{x}}_O(s) = \mathcal{F}(\hat{\boldsymbol{q}}_i(s)) - \begin{bmatrix} {}^I\boldsymbol{R}_O \boldsymbol{l}_i \\ \boldsymbol{\alpha}_i \end{bmatrix}, \;\; s \in [t_j, t_j + T_P], \tag{26d}$$

$$\hat{\boldsymbol{v}}_O(s) = \boldsymbol{J}_{i_O}\boldsymbol{J}_i(\hat{\boldsymbol{q}}_i(s))\dot{\hat{\boldsymbol{q}}}_i(s), \qquad s \in [t_j, t_j + T_P], \tag{26e}$$

$$\hat{\boldsymbol{x}}_i(s) \in X_i, \qquad s \in [t_j, t_j + T_P], \tag{26f}$$

$$\hat{\boldsymbol{\tau}}_i(s) \in T_i, \qquad s \in [t_j, t_j + T_P], \tag{26g}$$

where F and E are the running and terminal cost function respectively which are both of quadratic form i.e., $F(\cdot) = \hat{\boldsymbol{x}}_O^\top \boldsymbol{Q}_x \hat{\boldsymbol{x}}_O + \hat{\boldsymbol{v}}_O^\top \boldsymbol{Q}_v \hat{\boldsymbol{v}}_O + \boldsymbol{\tau}_i{}^\top \boldsymbol{R} \boldsymbol{\tau}_i$ and $E(\cdot) = \hat{\boldsymbol{x}}_O^\top \boldsymbol{P}_x \hat{\boldsymbol{x}}_O$, respectively, with $\boldsymbol{P}_x$, $\boldsymbol{Q}_x$, $\boldsymbol{Q}_v$ and $\boldsymbol{R}$ being positive definite matrices to be appropriately tuned [36]. In order to distinguish the predicted variables (i.e., internal to the controller) we use the double subscript notation $(\hat{\cdot})$ corresponding to the system (26b). This means that $\hat{\boldsymbol{x}}_i(s), \; s \in [t_j, t_j + T_P]$ is the solution of (18) based on the measurement of the state at time instance t_j (i.e., $\boldsymbol{x}_i(t_j)$) while applying a trajectory of inputs (i.e., $\hat{\boldsymbol{u}}_i(s), \; s \in [t_j, t_j + T_P]$). The solution of FHOCP (26a)-(26g) at time t_j provides an optimal control input trajectory denoted by $\hat{\boldsymbol{\tau}}_i^*(s; \boldsymbol{x}(t_j)), \; s \in [t_j, t_j + T_P]$. This control input is then applied to the system until the next sampling time t_{j+1}: i.e., $\boldsymbol{\tau}_i(s; \boldsymbol{x}(t_j)) = \hat{\boldsymbol{\tau}}_i^*(s; \boldsymbol{x}(t_j)), \; s \in [t_j, t_j + h]$. At time $t_{j+1} = t_j + h$ a new finite horizon optimal control problem is solved in the same manner, leading to a receding horizon approach. Notice that the control input $\boldsymbol{\tau}_i(\cdot)$ is of feedback form, since it is recalculated at each sampling instant based on the then-current state.

IV. Simulation Study

Real-time simulation have been performed to validate the proposed approach. The simulation environment is designed based on UwSim dynamic simulator [37] running on the Robot Operating System (ROS) [38]. We consider a scenario involving 3D motion with two UVMSs with the same structure, transporting a bar in a constrained workspace with static obstacles (see Fig.1). The UVMS model is an AUV equipped with a small 4 DoF manipulator attached at the bow of the vehicle (see Fig.1). The dynamic parameters of the vehicle have been identified via a proper identification scheme [39], while the manipulator's parameters as well as object's parameters have been extrapolated by the CAD data. The complete state vector of the vehicle (3D position, orientation, velocity) is available via the sensor fusion and state estimation module given in our previous results [39]. The Constrained NMPC employed in this work is implemented using the NLopt Optimization library [40].

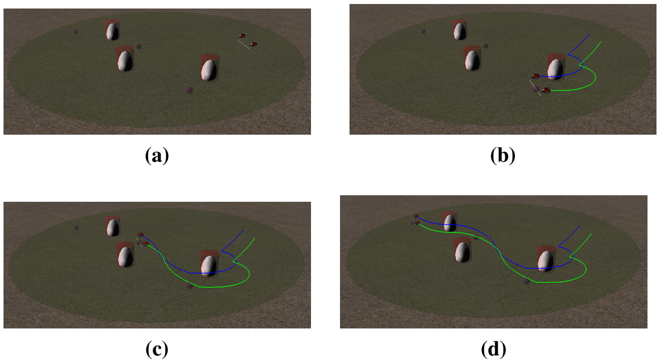

Fig. 3: The evolution of the proposed methodology in 4 consecutive time instants.

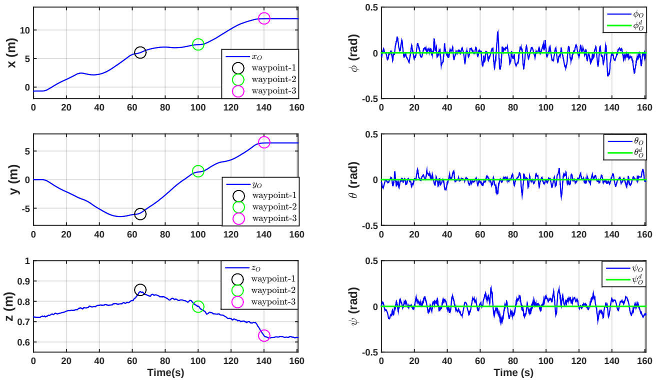

Fig. 4: Object coordinates during the control operation

A. Simulation results

In the following simulation, the objective for the team of UVMSs is to follow a set of predefined way points, while simultaneously avoid obstacles within the workspace. The position of the obstacles w.r.t the inertial frame $\mathcal{I}$ in $x - y$ plane is given by: $\boldsymbol{x}_{obs_1} = [4, \; -4.5]$, $\boldsymbol{x}_{obs_2} = [9, \; -1.5]$ and $\boldsymbol{x}_{obs_3} = [9, \; 5]$ respectively. These obstacles are cylinders with radius $r_{\pi_i} = 0.6m, \; i = \{1, 2, 3\}$ and are modeled together with the workspace boundaries according to the spherical world representations as consecutive spheres. The radius of the sphere $\mathcal{B}(\boldsymbol{p}_i, \bar{r}), \; i \in \{1, 2\}$ which covers all the UVMS volume (for all possible configurations) is defined as $\bar{r} = 1m$. In this way, the Navigation function (19)-(20) was designed with gain $K_{NF} = 0.5$. Regarding to constraints (23c), we consider that the vehicle's velocity most not exceed $0.5m/s$for translation and $0.1rad/s$ for rotational. In the same vein, the manipulator joint velocities

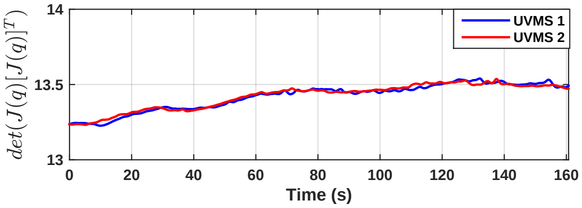

Fig. 5: $det(\boldsymbol{J}(\boldsymbol{q})[\boldsymbol{J}(\boldsymbol{q})]^\top)$ during the control operation

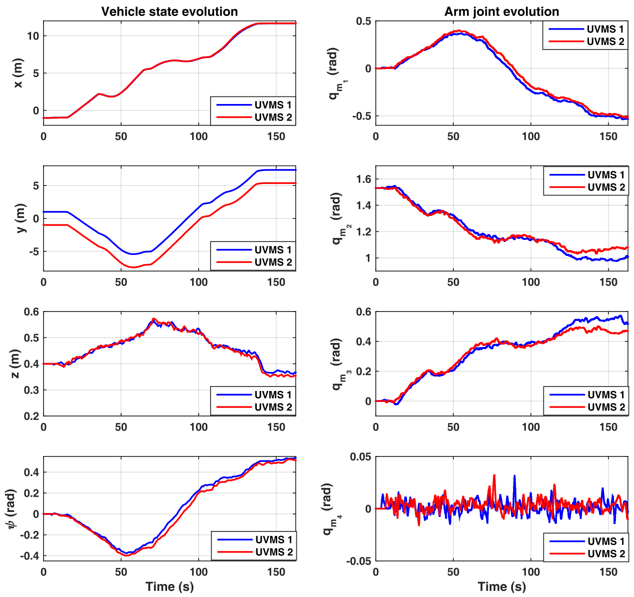

Fig. 6: The evolution of the system states at joint level

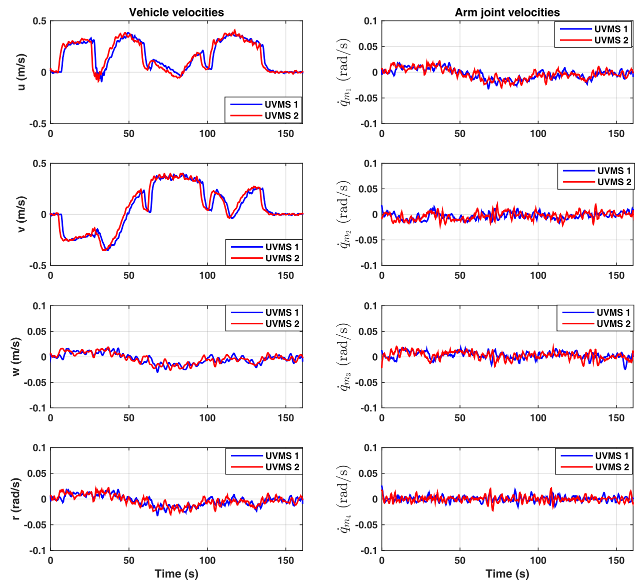

Fig. 8: The evolution of the system velocities at joint level

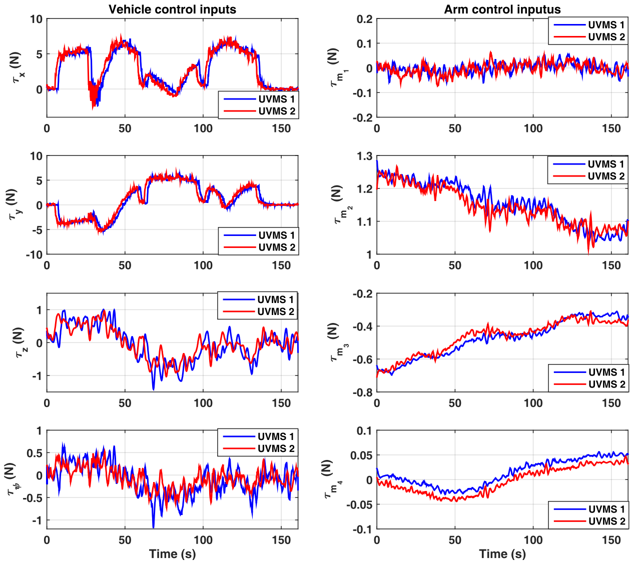

Fig. 7: The control input signals during the control operation

must be retained between $(-0.1, 0.1)rad/s$. Moreover, the manipulator joint positions (24) must be retained between $(-2, 2)rad$. Furthermore, input saturations (25) for the vehicle and manipulator are considered as: $\bar{\tau}_v = 10N$ and $\bar{\tau}_m = 2N$, respectively. The sampling time (21) and the prediction horizon are $h = 0.12sec$ and $T_p = 5 \times h = 0.6sec$ respectively. The matrices $\boldsymbol{P}_x$, $\boldsymbol{Q}_x$, $\boldsymbol{Q}_v$ and $\boldsymbol{R}$ as well as he load sharing coefficients c_1 and c_2 for both UVMSs are equal and set to: $\boldsymbol{P}_x = \boldsymbol{Q}_x = 0.8 \cdot \boldsymbol{I}_{6\times6}$, $\boldsymbol{R} = 0.3 \cdot \boldsymbol{I}_{8\times8}$, $\boldsymbol{Q}_v = 0.4 \cdot \boldsymbol{I}_{6\times6}$, and $c_1 = c_2 = 0.5$. The initial position of the object is $\boldsymbol{x}_O = [-0.7,\ 0,\ 0.72,\ 0.04,\ -0.07,\ 0]$. We set 3 waypoints as $\boldsymbol{x}^d_{O_1} = [6,\ -6,\ 0.85,\ 0,\ 0,\ 0]$, $\boldsymbol{x}^d_{O_2} = [7.5,\ 1.5,\ 0.78,\ 0,\ 0,\ 0]$ and $\boldsymbol{x}^d_{O_3} = [12,\ 6.5,\ 0.65,\ 0,\ 0,\ 0]$ which make the mission more challenging considering the obstacles' positions within the workspace (See Fig.3 and Fig.1). The results are presented in Fig.3-Fig.5. The trajectory of the system within the workspace as well as object coordinates evolution are depicted in Fig. 3 and Fig.4 respectively. It can be seen that the UVMSs have successfully transported cooperatively the object and have followed the set of predefined way points while safely avoids the obstacles. The evolution of $det(\boldsymbol{J}(\boldsymbol{q})[\boldsymbol{J}(\boldsymbol{q})]^\top)$ (see (3) and (23b)) during the operation is given in Fig.5. It can be easily seen that value remained positive during the cooperative manipulation task. Moreover, the evolution of the system velocity and its states at joints level as well as the corresponding control inputs are indicated in Fig.8, Fig. 6 and Fig.7 respectively. As it was expected from the theoretical findings, these values were retained in the corresponding feasible regions defined by the corresponding upper bounds and consequently all of the system constraints were satisfied.

V. Summary and Future Work

In this paper we presented a novel distributed object transportation control scheme for a team UVMSs in a constrained workspace with static obstacles. Various limitation and constraints such as: obstacles, joint limits, control input saturation as well as kinematic and representation singularities have been considered during the control design. The proposed control strategy relieves the team of robots from intense inter-robot communication during the execution of the collaborative tasks. This, avoids any restrictions imposed by the acoustic communication bandwidth (e.g., the number of participating UVMSs). Moreover, the control scheme adopts load sharing among the UVMSs according to their specific payload capabilities. Future research efforts will be devoted towards experimental validation of the proposed methodology with two small UVMSs inside a small test tank. In the same spirit, considering non-holonomic constraints on the UVMS model is a promising direction that would increase the applicability of the proposed control scheme.

References

[1] S. Heshmati-Alamdari, A. Eqtami, G. C. Karras, D. V. Dimarogonas, and K. J. Kyriakopoulos, "A self-triggered visual servoing model predictive control scheme for under-actuated underwater robotic vehicles," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3826–3831, IEEE, 2014.

[2] C. P. Bechlioulis, G. C. Karras, S. Heshmati-Alamdari, and K. J. Kyriakopoulos, "Trajectory tracking with prescribed performance for underactuated underwater vehicles under model uncertainties and external disturbances," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 429–440, 2017.

[3] P. Ridao, M. Carreras, D. Ribas, P. Sanz, and G. Oliver, "Intervention auvs: The next challenge," *Annual Reviews in Control*, vol. 40, pp. 227–241, 2015.

[4] S. Heshmati-alamdari, A. Nikou, K. Kyriakopoulos, and D. Dimarogonas, "A robust force control approach for underwater vehicle manipulator systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11197–11202, 2017.

[5] H. Farivarnejad and S. Moosavian, "Multiple impedance control for object manipulation by a dual arm underwater vehicle-manipulator system," *Ocean Engineering*, vol. 89, pp. 82–98, 2014.

[6] G. Marani, S. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions auvs," *Ocean Engineering*, vol. 36, no. 1, pp. 15–23, 2009.

[7] S. Heshmati-Alamdari, C. P. Bechlioulis, G. C. Karras, A. Nikou, D. V. Dimarogonas, and K. J. Kyriakopoulos, "A robust interaction control approach for underwater vehicle manipulator systems," *Annual Reviews in Control*, vol. 46, pp. 315–325, 2018.

[8] J. Fernández, M. Prats, P. Sanz, J. García, R. Marín, M. Robinson, D. Ribas, and P. Ridao, "Grasping for the seabed: Developing a new underwater robot arm for shallow-water intervention," *IEEE Robotics and Automation Magazine*, vol. 20, no. 4, pp. 121–130, 2013.

[9] E. Simetti, G. Casalino, S. Torelli, A. Sperindé, and A. Turetta, "Floating underwater manipulation: Developed control methodology and experimental validation within the trident project," *Journal of Field Robotics*, vol. 31, no. 3, pp. 364–385, 2014.

[10] M. Prats, D. Ribas, N. Palomeras, J. García, V. Nannen, S. Wirth, J. Fernández, J. Beltrán, R. Campos, P. Ridao, P. Sanz, G. Oliver, M. Carreras, N. Gracias, R. Marín, and A. Ortiz, "Reconfigurable auv for intervention missions: A case study on underwater object recovery," *Intelligent Service Robotics*, vol. 5, no. 1, pp. 19–31, 2012.

[11] D. Ribas, P. Ridao, A. Turetta, C. Melchiorri, G. Palli, J. Fernandez, and P. Sanz, "I-auv mechatronics integration for the trident fp7 project," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2583–2592, 2015.

[12] A. Carrera, N. Palomeras, D. Ribas, P. Kormushev, and M. Carreras, "An intervention-auv learns how to perform an underwater valve turning," *OCEANS 2014 - TAIPEI*, 2014.

[13] N. Hurtos, N. Palomeras, A. Carrera, M. Carreras, C. P. Bechlioulis, G. C. Karras, S. Heshmati-alamdari, and K. Kyriakopoulos, "Sonar-based chain following using an autonomous underwater vehicle," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1978–1983, IEEE, 2014.

[14] J. Gancet, D. Urbina, P. Letier, M. Ilzokvitz, P. Weiss, F. Gauch, G. Antonelli, G. Indiveri, G. Casalino, A. Birk, M. Pfingsthorn, S. Calinon, A. Tanwani, A. Turetta, C. Walen, and L. Guilpain, "Dexrov: Dexterous undersea inspection and maintenance in presence of communication latencies," *IFAC-PapersOnLine*, vol. 28, no. 2, pp. 218–223, 2015.

[15] R. Cui, S. Ge, B. Voon Ee How, and Y. Sang Choo, "Leader-follower formation control of underactuated autonomous underwater vehicles," *Ocean Engineering*, vol. 37, no. 17-18, pp. 1491–1502, 2010.

[16] D. J. Stilwell and B. E. Bishop, "Framework for decentralized control of autonomous vehicles," *In Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2358–2363, 2000.

[17] S. Heshmati-Alamdari, C. Bechlioulis, G. Karras, and K. Kyriakopoulos, "Decentralized impedance control for cooperative manipulation of multiple underwater vehicle manipulator systems under lean communication," *IEEE OES Autonomous Underwater Vehicle Symposium*, 2018.

[18] A. Nikou, C. Verginis, S. Heshmati-Alamdari, and D. V. Dimarogonas, "A nonlinear model predictive control scheme for cooperative manipulation with singularity and collision avoidance," in *2017 25th Mediterranean Conference on Control and Automation (MED)*, pp. 707–712, IEEE, 2017.

[19] C. K. Verginis, A. Nikou, and D. V. Dimarogonas, "Communication-based decentralized cooperative object transportation using nonlinear model predictive control," in *2018 European Control Conference (ECC)*, pp. 733–738, IEEE, 2018.

[20] R. Conti, E. Meli, A. Ridolfi, and B. Allotta, "An innovative decentralized strategy for i-auvs cooperative manipulation tasks," *Robotics and Autonomous Systems*, vol. 72, pp. 261–276, 2015.

[21] R. Furferi, R. Conti, E. Meli, and A. Ridolfi, "Optimization of potential field method parameters through networks for swarm cooperative manipulation tasks," *International Journal of Advanced Robotic Systems*, vol. 13, no. 5, pp. 1–13, 2016.

[22] E. Simetti and G. Casalino, "Manipulation and transportation with cooperative underwater vehicle manipulator systems," *IEEE Journal of Oceanic Engineering*, 2016.

[23] N. Manerikar, G. Casalino, E. Simetti, S. Torelli, and A. Sperinde, "On cooperation between autonomous underwater floating manipulation systems," *2015 IEEE Underwater Technology, UT 2015*, 2015.

[24] N. Manerikar, G. Casalino, E. Simetti, S. Torelli, and A. Sperindé, "On autonomous cooperative underwater floating manipulation systems," *In Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2015-June, 2015.

[25] E. Simetti, G. Casalino, N. Manerikar, A. Sperindé, S. Torelli, and F. Wanderlingh, "Cooperation between autonomous underwater vehicle manipulations systems with minimal information exchange," in *OCEANS 2015-Genova*, pp. 1–6, IEEE, 2015.

[26] S. Schneider and J. Cannon, R.H., "Object impedance control for cooperative manipulation: Theory and experimental results," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 383–394, 1992.

[27] J. Gudiño-Lau, M. A. Arteaga, L. A. Munoz, and V. Parra-Vega, "On the control of cooperative robots without velocity measurements," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 600–608, 2004.

[28] F. Caccavale, P. Chiacchio, and S. Chiaverini, "Task-space regulation of cooperative manipulators," *Automatica*, vol. 36, no. 6, pp. 879–887, 2000.

[29] H. Farivarnejad and S. A. A. Moosavian, "Multiple impedance control for object manipulation by a dual arm underwater vehicle–manipulator system," *Ocean Engineering*, vol. 89, pp. 82–98, 2014.

[30] S. A. A. Moosavian and E. Papadopoulos, "Multiple impedance control for object manipulation," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 1, pp. 461–466, IEEE, 1998.

[31] S. A. A. Moosavian and E. Papadopoulos, "Cooperative object manipulation with contact impact using multiple impedance control," *International Journal of Control, Automation and Systems*, vol. 8, no. 2, pp. 314–327, 2010.

[32] F. Allgöwer, R. Findeisen, and Z. Nagy, "Nonlinear model predictive control: From theory to application," *the Chinese Institute of Chemical Engineers*, vol. 35, no. 3, pp. 299–315, 2004.

[33] D. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Advances in Applied Mathematics*, vol. 11, no. 4, pp. 412–442, 1990.

[34] G. Antonelli, *"Underwater Robots"*. Springer Tracts in Advanced Robotics, Springer International Publishing, 2013.

[35] B. Siciliano, L. Sciavicco, and L. Villani, *Robotics : modelling, planning and control*. Advanced Textbooks in Control and Signal Processing, Springer, 2009. 013-81159.

[36] S. Heshmati-alamdari, G. C. Karras, P. Marantos, and K. J. Kyriakopoulos, "A robust model predictive control approach for autonomous underwater vehicles operating in a constrained workspace," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–5, 2018.

[37] M. Prats, J. Perez, J. Fernandez, and P. Sanz, "An open source tool for simulation and supervision of underwater intervention missions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2577–2582, 2012.

[38] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," *ICRA Workshop on Open Source Software*, 2009.

[39] G. C. Karras, P. Marantos, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Unsupervised online system identification for underwater robotic vehicles," *IEEE Journal of Oceanic Engineering*, 2018.

[40] S. G. Johnson, "The nlopt nonlinear-optimization package," *http://ab-initio.mit.edu/nlopt*.