A Predictive Control Approach for Cooperative Transportation by 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 saturation. More precisely, by exploiting the coupled dynamics between the robots and the object, and using certain load sharing coefficients, we design a predictive controller for each UVMS in order to cooperatively transport the object within the workspaces feasible region. Moreover, the control scheme adopts load sharing among the UVMSs according to their specific payload capabilities. Additionally, the feedback relies on each UVMSs on-board measurements and no explicit data is exchanged online among the robots, thus reducing the required communication bandwidth. Finally, realistic simulation results conducted in UwSim dynamic simulator running in ROS environment as well as real time experiments employing two small UVMSs, demonstrated the effectiveness of the proposed control strategy.**

***Index Terms*—Underwater Vehicle Manipulator System, Cooperative Manipulation, Marine Robotics, Nonlinear Model Predictive Control, Underwater Navigation and Control.**

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) and offshore industry (e.g., ship maintenance, inspection of oil/gas facilities) [1]. In particular, a vast number of the aforementioned applications, demand the underwater vehicle to be enhanced with intervention capabilities [2], hence raising the interest on Underwater Vehicle Manipulator System (UVMS) [3]. Nowadays, underwater intervention tasks involve a Remotely Operated Vehicle (ROV), equipped with one or multiple manipulators that allow it to grasp, transport and manipulate objects while being controlled by a human pilot on a surface ship [4]. However, the well-known disadvantages of human-robot tele-operation led inevitably to the development of autonomous intervention control schemes for UVMS that have gained significant scientific attention during the last years [5].

More specifically, during late 90s, early efforts towards designing underwater vehicle manipulator systems were made within the pioneering projects AMADEUS [6] and UNION [7]. A more recent European project, which has boosted autonomous underwater interaction tasks, was the TRIDENT [8], where a vehicle-arm system was controlled in a coordinated manner. Another important milestone was achieved within the PANDORA project [9], where a strong emphasis was given on the problem of persistent autonomy. Finally, the most recent project in the domain of underwater intervention was the DexROV project [10], which focused on inspection and maintenance tasks via satellite communications in the presence of latencies.

Most of the underwater manipulation tasks can be carried out more efficiently, if multiple UVMSs are cooperatively involved. However, underwater multi-robot tasks are very demanding, with the most significant challenge being imposed by the strict communication constraints [11]. In general, the communication of multi-robot systems can be classified in two major categories, namely explicit and implicit. The most frequently employed communication form in multi-robot systems is the explicit one. However, it's employing in an 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 robots in this case, is strictly limited owing to the narrow bandwidth of acoustic communication devices [12]. To overcome such limitations, recent studies on underwater cooperative manipulation are dealing with designing control schemes under lean communication requirements.

Cooperative manipulation has been well-studied in the literature, especially the centralized schemes. On the other hand, although decentralized cooperative manipulation schemes exhibit increased robustness and low complexity, they usually depend on explicit communication interchange among the robots. For instance, in recent studies [13], [14], 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 towards the same direction have been given in [15], [16] 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 team, which consequently restricts the number of robots involved in the cooperative manipulation task.

Moreover, various studies can be found in the literature employing decentralized cooperative manipulation control schemes where robotic agents use only their local information or observation [17]. 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

Shahab Heshmati-alamdari is with the Section of Automation & Control, Department of Electronic Systems, Aalborg University, Denmark. G. C. Karras is with University of Thessaly, Lamia, Greece and with the Control Systems Lab, School of Mechanical Engineering, National Technical University of Athens, Greece. K. J. Kyriakopoulos are with the Control Systems Lab, School of Mechanical Engineering, National Technical University of Athens, Greece. Email: shhe@se.aau.dk, {karrasg,kkyria@mail.ntua.gr}.

reduction due to sensor noise [18], [19]. 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 [1], the problem of decentralized cooperative object transportation considering multiple UVMSs in a constrained workspace with static obstacles is addressed. Specifically, given N UVMSs rigidly grasp a common object, we design decentralized 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 Nonlinear Model Predictive Control (NMPC) [21], [22] 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 [23] 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 the 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. In addition, it should be noticed that in the proposed methodology, each UVMS calculates its control signals without communicating with each other and exploits information acquired solely by onboard sensors, i.e., position and velocity measurements (e.g., sensor fusion based on measurement of various onboard sensors such as IMU, USBL and DVL), avoiding thus any tedious inter-robot explicit communication. 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). Finally, to the best of the authors knowledge and compared to the existing works in the literature, this is the first time where a cooperative manipulation scheme for multiple Underwater Vehicle Manipulator Systems is experimentally verified. Furthermore, it should be noticed that the proposed control strategy is more complete with respect to existing works, since it incorporates input (i.e., thrust saturation) and state (e.g., kinematic and representation singularities, 3D obstacles within the workspace, joint limits) constraints into the system's closed-loop motion while avoids any restrictions imposed by the strict underwater communication bandwidth.

The rest of the paper is organized as follows: In Section-II, the mathematical modeling along with the verbal description of the problem statement are presented. An analytical description of the proposed control method is presented in Section-III. The efficiency of the proposed approach is illustrated in Section-IV via both simulation and experimental studies. Finally, Section-V concludes the paper.

II. Problem Formulation

Consider N UVMSs rigidly grasping an object[2] within a constrained workspace with static obstacles. We also assume that each UVMS is fully-actuated at its end-effector frame. This assumption implies that all UVMSs are able to exert arbitrary forces and torques on the object along and about any direction. Moreover, we assume that the UVMSs are equipped with appropriate sensors, that allow them to measure their position and velocity. Additionally, the geometric parameters of the both UVMSs and the commonly grasped object are considered known.

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 with respect to 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 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 base and $\boldsymbol{q}_{m,i}$ is the vector of the angular positions of the manipulator's joints. More 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 with respect to the inertial frame. Thus, we have [1], [24]:

$$\dot{\boldsymbol{q}}_{B,i} = \boldsymbol{J}_{B,i}(\boldsymbol{q}_{B,i})\boldsymbol{\rho}_i,\ i \in \mathcal{N} \tag{1}$$

where $\boldsymbol{\rho}_i$ is the velocity of the vehicle expressed in the body-fixed frame and $\boldsymbol{J}_{B,i}(\boldsymbol{q}_{v,i})$ is the Jacobian matrix transforming the velocities from the body-fixed to 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 with respect to inertial frame, is given by the forward kinematics of the complete system (arm and vehicle base) as follows:

$$\boldsymbol{p}_i = \mathcal{F}(\boldsymbol{q}_i)\ , i \in \mathcal{N} \tag{2}$$

Moreover, without any loss of generality, for the augmented UVMS system we get [24]:

$$\boldsymbol{v}_i = \boldsymbol{J}_i(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i,\ i \in \mathcal{N} \tag{3}$$

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 with respect to the inertial frame as well as the joint velocities of the manipulator and $\boldsymbol{J}_i(\boldsymbol{q}_i)$ is the geometric Jacobian matrix [24]. Note

[1] A preliminary version of this work, in the absence of a detailed analysis of the methodology, including detailed geometric modeling of the system as well as of the coupled dynamics, detailed controller design, description of real time implementation and a enriched set of experimental results using two small UVMSs was reported in [20].

[2] The end-effector frame of each UVMS is always constant relative to the object's body fixed frame.

that the $\boldsymbol{J}_{B,i}$ becomes singular at representation singularities, when $\theta_{B_i} = \pm\frac{\pi}{2}$ and $\boldsymbol{J}_i(\boldsymbol{q}_i)$ becomes singular when $\det(\boldsymbol{J}_i(\boldsymbol{q}_i)[\boldsymbol{J}_i(\boldsymbol{q}_i)]^\top = 0$, thus, we aim at guaranteeing that $\boldsymbol{q}_i$ will always be in the closed set:

$$Q_{s_i} = \{\boldsymbol{q}_i \in \mathbb{R}^{n_i} : \det(\boldsymbol{J}_i(\boldsymbol{q}_i)[\boldsymbol{J}_i(\boldsymbol{q}_i)]^\top) \geq \epsilon\},\ i \in \mathcal{N}. \quad (4)$$

with ϵ to be a small positive number.

B. UVMS Dynamics

Without any loss of generality, the dynamics of a UVMS can be written as [24]:

$$\boldsymbol{M}_{q_i}(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \boldsymbol{C}_{q_i}(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{D}_{q_i}(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{g}_{q_i}(\boldsymbol{q}_i) = \boldsymbol{\tau}_i - \boldsymbol{J}_i{}^\top \boldsymbol{\lambda}_i \quad (5)$$

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 (3) we have:

$$\dot{\boldsymbol{v}}_i = \boldsymbol{J}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \dot{\boldsymbol{J}}_i(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i,\ i \in \mathcal{N} \quad (6)$$

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 (3) as well as (6), we obtain from (5) the transformed task space dynamics [25]:

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

for all $i \in \mathcal{N}$ with corresponding task space terms $\boldsymbol{M}_i \in \mathbb{R}^{6\times6}$, $\boldsymbol{C}_i \in \mathbb{R}^{6\times6}$, $\boldsymbol{D}_i \in \mathbb{R}^{6\times6}$, $\boldsymbol{g}_i \in \mathbb{R}^6$:

$$\begin{gathered}
\boldsymbol{M}_i(\boldsymbol{q}_i) = [\boldsymbol{J}_i(\boldsymbol{q}_i)\boldsymbol{M}_{q_i}^{-1}\boldsymbol{J}_i(\boldsymbol{q}_i)^\top]^{-1} \\
\boldsymbol{C}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\boldsymbol{J}_i(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i = \boldsymbol{M}_i(\boldsymbol{q}_i)\big[\boldsymbol{J}_i(\boldsymbol{q}_i)\boldsymbol{M}_{q_i}^{-1}\boldsymbol{C}_{q_i} - \dot{\boldsymbol{J}}_i(\boldsymbol{q}_i)\big]\dot{\boldsymbol{q}}_i \\
\boldsymbol{D}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\boldsymbol{J}_i(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i = \boldsymbol{M}_i(\boldsymbol{q}_i)\boldsymbol{J}_i(\boldsymbol{q}_i)\boldsymbol{M}_{q_i}^{-1}\boldsymbol{D}_{q_i}\dot{\boldsymbol{q}}_i \\
\boldsymbol{g}_i(\boldsymbol{q}_i) = \boldsymbol{M}_i(\boldsymbol{q}_i)\boldsymbol{J}_i(\boldsymbol{q}_i)\boldsymbol{M}_{q_i}^{-1}\boldsymbol{g}_{q_i}
\end{gathered}$$

Moreover, $\boldsymbol{u}_i \in \mathbb{R}^6$ is the vector of task space generalized forces/torques. It is worth noting that the vector of control inputs $\boldsymbol{\tau}_i,\ i \in \mathcal{K}$ can be related to the task space wrench $\boldsymbol{u}_i \in \mathbb{R}^6,\ i \in \mathcal{K}$ via:

$$\boldsymbol{\tau}_i = \boldsymbol{J}_i^\top(\boldsymbol{q}_i)\boldsymbol{u}_i + \boldsymbol{\tau}_{i0}(\boldsymbol{q}_i) \quad (8)$$

where the vector $\boldsymbol{\tau}_{i0}(\boldsymbol{q}_i)$ does not contribute to the end effectors wrench $\boldsymbol{u}_i$ (i.e., it belongs to the null space of the Jacobian $\boldsymbol{J}_i^\top$) and can be regulated independently to achieve secondary tasks (e.g., maintaining manipulator's joint limits, increasing the manipulability) [25], [26]. The UVMS task space dynamics (7) can be written in vector form as:

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}} + \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\boldsymbol{v} + \boldsymbol{D}(\dot{\boldsymbol{q}}, \boldsymbol{q})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u} - \boldsymbol{\lambda} \quad (9)$$

where $\boldsymbol{v} = [\boldsymbol{v}_1^\top, \ldots, \boldsymbol{v}_N^\top]^\top \in \mathbb{R}^{6N}$, $\boldsymbol{M} = \text{diag}\{[\boldsymbol{M}_i]\} \in \mathbb{R}^{6N\times6N}$, $\boldsymbol{C} = \text{diag}\{[\boldsymbol{C}_i]\} \in \mathbb{R}^{6N\times6N}$, $\boldsymbol{D} = \text{diag}\{[\boldsymbol{D}_i]\} \in \mathbb{R}^{6N\times6N}$, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^\top, \ldots, \boldsymbol{\lambda}_N^\top]^\top$, $\boldsymbol{u} = [\boldsymbol{u}_1^\top, \ldots, \boldsymbol{u}_N^\top]^\top$, $\boldsymbol{g} = [\boldsymbol{g}_1^\top, \ldots, \boldsymbol{g}_N^\top]^\top \in \mathbb{R}^{6N}$.

C. Object Dynamics

We denote the coordinates of the object by $\boldsymbol{x}_O = [\boldsymbol{\eta}_{1,O}^\top, \boldsymbol{\eta}_{2,O}^\top]^\top$ where $\boldsymbol{\eta}_{1,O}^\top = [x_O, y_O, z_O]^\top$ and $\boldsymbol{\eta}_{2,O}^\top = [\phi_O, \theta_O, \psi_O]^\top$ denote the position and the orientation expressed in Euler angles with respect to the inertial frame. Let also define the object generalized velocities by $\boldsymbol{v}_O = [\dot{\boldsymbol{\eta}}_{1,O}^\top, \boldsymbol{\omega}_O^\top]^\top$. Without any loss of generality, we consider the following second order dynamics for the object, which can be derived based on the Newton-Euler formulations:

$$\dot{\boldsymbol{x}}_O = \boldsymbol{J}_O'(\boldsymbol{\eta}_{2,O})^{-1}\boldsymbol{v}_O \quad (10a)$$

$$\boldsymbol{M}_O(\boldsymbol{x}_O)\dot{\boldsymbol{v}}_O + \boldsymbol{C}_O(\boldsymbol{v}_O, \boldsymbol{x}_O)\boldsymbol{v}_O + \boldsymbol{D}_O(\boldsymbol{v}_O, \boldsymbol{x}_O)\boldsymbol{v}_O + \boldsymbol{g}_O = \boldsymbol{\lambda}_O \quad (10b)$$

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 can be given as:

$$\boldsymbol{J}_O'(\boldsymbol{\eta}_{2,O}) = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{J}_O''(\boldsymbol{\eta}_{2,O}) \end{bmatrix}, \quad (11)$$

with: $\boldsymbol{J}_O''(\boldsymbol{\eta}_{2,O}) = \begin{bmatrix} 1 & 0 & -\sin(\theta_O) \\ 0 & \cos(\phi_O) & \cos(\theta_O)\sin(\phi_O) \\ 0 & -\sin(\phi_O) & \cos(\theta_O)\cos(\phi_O) \end{bmatrix}$. Note that the $\boldsymbol{J}_O'(\boldsymbol{\eta}_{2,O})$ is singular when $\theta_O = \pm\frac{\pi}{2}$ [24].

D. Problem statement

Herein, we address the problem under consideration:

Problem 1: Given i) N UVMSs operating in a constrained workspace $\mathcal{W}$ and rigidly grasping an object as well as ii) a desired configuration for the object $\boldsymbol{x}_O^d$, design distributed control protocols $\boldsymbol{\tau}_i,\ \in \mathcal{N}$ that navigate safely the whole robotic team to the desired configuration while satisfying the following specifications:

1) Impose no strict requirements regarding the underwater communication bandwidth;
2) Avoiding constraints and limitations such as: kinematic and representation singularities, joint limits and control input saturation;
3) Collision avoidance with the obstacles and the boundary of the workspace;
4) Achieve distributed consensus on a mutually agreed trajectory of the commonly grasped object;
5) Adopting load sharing among the UVMSs according to their specific payload capabilities;
6) The feedback relies on each UVMSs locally measurements and no explicit data is exchanged online among the robots.

III. Control Methodology

The control objective is to navigate of the overall formation towards the goal configuration while avoiding collisions with the static obstacles that lie within the workspace. First, the overall dynamics of the system (i.e., object and robots) are formulated accordingly. We achieve a decoupled form of the system dynamics by employing certain load coefficients. Each UVMS at each sampling time, solves a NMPC subject to its corresponding part of the 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 guide cooperatively the object and steer it along a computed feasible path within the workspace. The feasible path computation is based on the concept of Navigation Functions [23] that is incorporated in order to deal with consensus on a mutually agreed trajectory of the commonly object.

Assumption 1: We assume that each UVMS $i \in \mathcal{N}$ is equipped with appropriate sensors that enable it to continuously measure its own state vector $\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i$ $i \in \mathcal{N}$ based on its own state measurements (by employing sensor fusion of locally onboard navigation system sensors, e.g., DVL, IMU, USBL and depth-sensor).

A. Coupled Dynamics

Consider the N UVMS rigidly grasping a common object. Owing to the rigid grasp of the object, the following equations hold:

$$\boldsymbol{p}_i = \boldsymbol{x}_O + \begin{bmatrix} {}^I\boldsymbol{R}_O \boldsymbol{l}_i \\ \boldsymbol{\alpha}_i \end{bmatrix}, \ i \in \mathcal{N} \tag{12}$$

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 which describes the orientation of the object expressed in the inertial frame $\{I\}$. Thus, using (12) 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 rigid grasp, it holds that $\boldsymbol{\omega}_i = \boldsymbol{\omega}_O, \ i \in \mathcal{N}$, hence we obtain:

$$\boldsymbol{v}_O = \boldsymbol{J}_{i_O} \boldsymbol{v}_i, \ i \in \mathcal{N} \tag{13}$$

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 $\boldsymbol{l}_i = [l_{ix}, l_{iy}, l_{iz}]^\top$ defined as:

$$\boldsymbol{S}(\boldsymbol{l}_i) = \begin{bmatrix} 0 & -l_{iz} & l_{iy} \\ l_{iz} & 0 & -l_{ix} \\ -l_{iy} & l_{ix} & 0 \end{bmatrix} \in \mathbb{R}^{3\times3}, \ i \in \mathcal{N}$$

Notice that $\boldsymbol{J}_{i_O}, \ i \in \mathcal{N}$ are always full-rank owing to the grasp rigidity and hence a well defined matrix inverse is obtained. Thus, the object's velocity can be easily computed via (13). Moreover, from (13), one obtains the acceleration relation:

$$\dot{\boldsymbol{v}}_O = \boldsymbol{J}_{i_O} \dot{\boldsymbol{v}}_i + \dot{\boldsymbol{J}}_{i_O} \boldsymbol{v}_i, \ i \in \mathcal{N} \tag{14}$$

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:

$$\boldsymbol{\lambda}_O = \boldsymbol{G}^\top \boldsymbol{\lambda} \tag{15}$$

where:

$$\boldsymbol{G} = \Big[[\boldsymbol{J}_{O_1}]^\top, \ldots, [\boldsymbol{J}_{O_N}]^\top \Big]^\top \in \mathbb{R}^{6N\times6} \tag{16}$$

is the full column-rank grasp matrix, $\boldsymbol{J}_{O_i} = [\boldsymbol{J}_{i_O}]^{-1}, \ i \in \mathcal{N}$ and $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^\top, \ldots, \boldsymbol{\lambda}_N^\top]^\top$ is the vector containing the overall interaction forces and torques.

Remark 1: Wrenches that lie on the null space of the grasp matrix $\boldsymbol{G}^\top$ do not contribute to the object dynamics. Therefore, we may incorporate in the control scheme an extra component $\boldsymbol{\lambda}_{int,i} = (\boldsymbol{I} - (\boldsymbol{G}^\top)^{\#} \boldsymbol{G}^\top) \boldsymbol{\lambda}_{int}^d, \ i \in \mathcal{N}$, that belongs to the null space of $\boldsymbol{G}^\top$, in order to regulate the steady state internal forces, where $(\boldsymbol{G}^\top)^{\#}$ denotes the generalized inverse of $\boldsymbol{G}^\top$. Notice that owing to the rigid grasp, $l_i, \ i \in \mathcal{N}$ remain constant. Thus, since $l_i, \ i \in \mathcal{N}$ are considered known to the team of UVMSs, if $\boldsymbol{\lambda}_{int}^d$ is chosen constant, no communication is needed during task execution in order to compute $\boldsymbol{G}^\top, \ (\boldsymbol{G}^\top)^{\#}$ and $\boldsymbol{\lambda}_{int,i}$.

By substituting (9) into (15) one obtains:

$$\boldsymbol{\lambda} = \boldsymbol{G}^\top \Big[\boldsymbol{u} - \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}} - \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\boldsymbol{v} - \boldsymbol{D}(\dot{\boldsymbol{q}}, \boldsymbol{q})\boldsymbol{v} - \boldsymbol{g}(\boldsymbol{q}) \Big] \tag{17}$$

which, after substituting (13), (14), (10) and rearranging terms, yields the overall system coupled dynamics:

$$\widetilde{\boldsymbol{M}}(\tilde{\boldsymbol{q}}_{ov})\dot{\boldsymbol{v}}_O + \widetilde{\boldsymbol{C}}(\tilde{\boldsymbol{q}}_{ov})\boldsymbol{v}_O + \widetilde{\boldsymbol{D}}(\tilde{\boldsymbol{q}}_{ov})\boldsymbol{v}_O + \widetilde{\boldsymbol{g}}(\tilde{\boldsymbol{q}}_{ov}) = \boldsymbol{G}^\top \boldsymbol{u} \tag{18}$$

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

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

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

$$c_i \in (0,1), \forall i \in \mathcal{N} \qquad \text{and} \qquad \sum_{i \in \mathcal{N}} c_i = 1, \tag{19}$$

which we introduce in order to act as the load sharing coefficients for the team of UVMS. In view of (19), the object dynamics (10b) can be rewritten as [27], [28]:

$$\sum_{i \in \mathcal{N}} c_i \Big\{ \boldsymbol{M}_O(\boldsymbol{x}_O)\dot{\boldsymbol{v}}_O + \boldsymbol{C}_O(\boldsymbol{x}_O, \boldsymbol{v}_O)\boldsymbol{v}_O + \boldsymbol{D}_O(\boldsymbol{x}_O, \boldsymbol{v}_O)\boldsymbol{v}_O + \boldsymbol{g}_O(\boldsymbol{x}_O) \Big\} = \sum_{i \in \mathcal{N}} \boldsymbol{J}_{O_i}^\top \boldsymbol{\lambda}_i \tag{20}$$

from which, by employing (3), (6), (13), (7) and (14), and after straightforward algebraic manipulations, we obtain the coupled dynamics:

$$\sum_{i \in \mathcal{N}} \Big\{ \widetilde{\boldsymbol{M}}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \widetilde{\boldsymbol{C}}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\dot{\boldsymbol{q}}_i + \widetilde{\boldsymbol{D}}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i)\dot{\boldsymbol{q}}_i + \widetilde{\boldsymbol{g}}_i(\boldsymbol{q}_i) \Big\} = \sum_{i \in \mathcal{N}} \boldsymbol{J}_{O_i}^\top \boldsymbol{u}_i \tag{21}$$

where:

$$\widetilde{\boldsymbol{M}}_i(\boldsymbol{q}_i) = c_i \boldsymbol{M}_O \boldsymbol{J}_{i_O} \boldsymbol{J}_i + \boldsymbol{J}_{O_i}^{\top} \boldsymbol{M}_i \boldsymbol{J}_i$$

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

$$\widetilde{\boldsymbol{D}}_i(\dot{\boldsymbol{q}}_i, \boldsymbol{q}_i) = c_i \boldsymbol{D}_O \boldsymbol{J}_{i_O} \boldsymbol{J}_i + \boldsymbol{J}_{O_i}^{\top} \boldsymbol{D}_i \boldsymbol{J}_i$$

$$\widetilde{\boldsymbol{g}}_i(\boldsymbol{q}_i) = c_i \boldsymbol{g}_O + \boldsymbol{J}_{O_i}^{\top} \boldsymbol{g}_i$$

where for each UVMS, it is based only 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 individual dynamics for each UVMS based on (21), can be given in compact form:

$$\dot{\boldsymbol{x}}_i = f_i(\boldsymbol{x}_i, \boldsymbol{u}_i) = \begin{bmatrix} f_{i_1}(\boldsymbol{x}_i) \\ f_{i_2}(\boldsymbol{x}_i, \boldsymbol{u}_i) \end{bmatrix}, i \in \mathcal{N} \tag{22}$$

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

Consider the team of N UVMSs operating in a bounded workspace $\mathcal{W} \subseteq \mathbb{R}^3$ with boundary $\partial\mathcal{W}$. The object of interest is a rigid body which is required to be transported cooperatively by the robot team from an initial to a goal position. Without any loss of the generality, the obstacles, the robots as well as the workspace are all modeled by spheres (i.e., we adopt the spherical world representation [23]). In this spirit, let $\mathcal{B}(\boldsymbol{x}_O, r_0)$ be a closed sphere that covers the volume of the object and has radius r_0. We also define the closed spheres $\mathcal{B}(\boldsymbol{p}_i, \bar{r}), i \in \mathcal{K}$, centered at the end-effector of each UVMS that cover the robot volume for all possible configurations. Notice that the value of $\bar{r}$ can be calculated easily for each UVMS based solely on its own design parameters. We also assume that the distance among the grasping points on the given object is at least $2\bar{r}$. In particular, 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. Furthermore, we define a sphere area $\mathcal{B}(\boldsymbol{x}_O, R)$ located at $\boldsymbol{x}_O$ with radius $R = \bar{r} + r_o$ that includes the complete volume of the robotic team and the object (see Fig. 1). Finally, the $\mathcal{M}$ static obstacles within the workspace 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 $r_{\pi_m} > 0$ the radius of the obstacle π_m. Obviously, the ultimate goal of the proposed cooperative control strategy is to transport the object from the initial configuration to the desired one, without colliding with the obstacles and the boundary of workspace. Therefore, based on the property of spherical world [23], for each pair of obstacles $m, m' \in \{1, \ldots, \mathcal{M}\}$ the following inequality holds:

$$||\boldsymbol{p}_{\pi_m} - \boldsymbol{p}_{\pi'_m}|| > 2R + r_{\pi_m} + r_{\pi'_m}$$

which intuitively means that the obstacles m and m' are disjoint in a such a way that the whole team of UVMSs including the object can pass through the free space between them. Therefore, there exists a feasible trajectory $\boldsymbol{x}_O(t)$ for the whole team that connects the initial configuration $\boldsymbol{x}_O(t_0)$ with $\boldsymbol{x}_O^d$ such as:

$$\mathcal{B}(\boldsymbol{x}_O(t), R) \cap \{\mathcal{B}(\boldsymbol{p}_{\pi_m}, r_{\pi_m}) \cup \partial\mathcal{W}\} = \emptyset,\ \forall m \in \{1, \ldots, \mathcal{M}\}$$

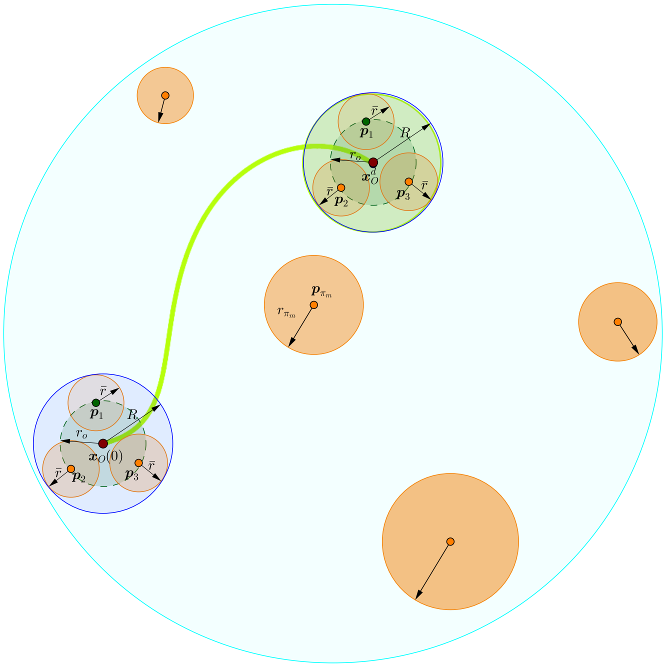


Fig. 1. Graphical representation of a safe trajectory of the robotic team. The boundary of workspace $\partial\mathcal{W}$ is illustrated in cyan. The orange areas indicate the obstacles within the workspace $\mathcal{W}$. The blue line encircles the area covered by the robotic team and the object. A safe trajectory of the whole team is depicted in green.

C. Safe Navigation

The desired object trajectory within the workspace $\mathcal{W}$ relies on the Navigation Function concept originally proposed by Rimon and Koditschek in [23] as follows:

$$\phi_O(\boldsymbol{x}_O; \boldsymbol{x}_O^d) = \frac{\gamma(\boldsymbol{x}_O - \boldsymbol{x}_O^d)}{[\gamma^k(\boldsymbol{x}_O - \boldsymbol{x}_O^d) + \beta(\boldsymbol{x}_O)]^{\frac{1}{k}}} \tag{23}$$

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 which should be selected sufficiently large in order to guarantee the navigation[3]. In addition, $\gamma(\boldsymbol{x}_O - \boldsymbol{x}_O^d) > 0$ with $\gamma(\boldsymbol{0}) = 0$

[3] For more details the reader is refer to [23][pg.428].

represents the attractive potential field to the goal configuration $\boldsymbol{x}_O^d$ and $\beta(\boldsymbol{x}_O) > 0$ with:

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

represents the repulsive potential field induced by the workspace boundary and the obstacle regions. In that respect, it was proven in [23] 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[4] 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:

$$\boldsymbol{v}_O^d(t) = -K_{NF} \boldsymbol{J}_O'(\boldsymbol{\eta}_{2,O}) \nabla_{\boldsymbol{x}_O} \phi_O(\boldsymbol{x}_O(t), \boldsymbol{x}_O^d) \tag{24}$$

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:

$$t_{j+1} = t_j + h, \ \forall j \geq 0 \tag{25}$$

Therefore, since all UVMS $i \in \mathcal{N}$ are aware of both the desired configuration of the object as well as of the obstacles position in the workspace, given a current position of the object $\boldsymbol{x}_O(t_i)$ and $\boldsymbol{v}_O(t_j)$ at the time t_j they can propagate 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 (23), (24) 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 assume that the UVMS must satisfy joint limits and singularity avoidance that can be considered as state constraints of the system. These requirements are captured by the state constraint set X_i of the system, given by:

$$\boldsymbol{x}_i(t) \in X_i \subset \mathbb{R}^{2n_i} \tag{26}$$

which is formed by the following constraints:

$$\theta_O(t) \in (-\frac{\pi}{2}, \frac{\pi}{2}) \tag{27}$$

$$\boldsymbol{q}_i \in \mathbb{R}^{n_i} \in \Big(Q_{s_i}(\boldsymbol{q}_i) \cup Q_{l_i}(\boldsymbol{q}_i)\Big), \ i \in \mathcal{N} \tag{28}$$

$$|\dot{q}_{k_i}| \leq \bar{\dot{q}}_{k_i}, \forall k \in \{1, \ldots, n\}, i \in \mathcal{N} \tag{29}$$

where $Q_{s_i}(\boldsymbol{q}_i)$ is the feasible set for the system (4) and $Q_{l_i}(\boldsymbol{q}_i)$ is the feasible set of manipulator's joint positions defined as:

$$Q_{l_i}(\boldsymbol{q}_i) = \{\boldsymbol{q}_i \in \mathbb{R}^{n_i} : |q_{k_i}| \leq \bar{q}_i\}, \forall k \in \{1, \ldots, n_i\}, i \in \mathcal{N} \tag{30}$$

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, $\bar{\dot{q}}_{k_i}$ 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 is a open and connected set that captures all the state constraints of the systems (22), i.e., singularity avoidance as well as joint limits.

Remark 2: Notice that collision avoidance between the complete system (UVMS and the object) and obstacles (see Fig 1) are achieved based on the desired trajectory and velocity of the object as calculated from (23) and (24).

Input Constraints:

The actuation of the vehicle and the manipulator are generated by the thrusters and servo motors respectively. Hence, the input constraints for $\tau_{k_i}, \ k \in \{1, \ldots, \tau_n\}, i \in \mathcal{N}$, with τ_n to be the number of actuated joints, can be given 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 the corresponding limit bound for each actuated joint $\tau_{k_i}, \ k \in \{1, \ldots, \tau_{n_i}\}, i \in \mathcal{N}$. Therefore, we can define the compact control input set T_i:

$$\boldsymbol{\tau}_i(t) \in T_i \subset \mathbb{R}^{\tau_{n_i}} \tag{31}$$

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

As it is already mentioned, given the current position and velocity of the object at sampling time j denoted by $\boldsymbol{x}_O(t_j)$ and $\boldsymbol{v}_O(t_j)$ respectively, each UVMS $i \in \mathcal{N}$ for a time interval $s \in [t_j, t_j + T_P]$ where T_P is a prediction horizon and based on (23), (24) and (25), can propagate a map of desired trajectory and velocity for the object denoted by $\boldsymbol{x}_O^d(s)$ and $\boldsymbol{v}_O^d(s)$ respectively. As it will be explained in the sequel, at each sampling time, UVMS $i \in \mathcal{N}$ solves its corresponding part of the dynamics (21) via an NMPC scheme subject to its dynamics (22) and a number of inequality constraints. More specifically, the control objective for each UVMS $i \in \mathcal{N}$ is to follow the desired trajectory and velocity, while respecting the state constraints (27)-(29) as well as the input constraints (31). 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 following FHOCP:

$$\min_{\hat{\boldsymbol{\tau}}_i(\cdot)} J_i(\boldsymbol{x}(t_j), \hat{\boldsymbol{\tau}}_i(\cdot)) = \tag{32a}$$

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

subject to:

$$\dot{\hat{\boldsymbol{x}}}_i(s) = f_i(\hat{\boldsymbol{x}}_i(s), \hat{\boldsymbol{u}}_i(s)), \quad \hat{\boldsymbol{x}}_i(t_j) = \boldsymbol{x}_i(t_j), \tag{32b}$$

$$\hat{\boldsymbol{\tau}}_i(s) = \boldsymbol{J}_i^\top(\hat{\boldsymbol{q}}_i)\hat{\boldsymbol{u}}_i + \boldsymbol{\tau}_{i0}(\boldsymbol{q}_i), \ s \in [t_j, t_j + T_P] \tag{32c}$$

$$\hat{\boldsymbol{x}}_O(s) = \mathcal{F}(\hat{\boldsymbol{q}}_i(s)) - \begin{bmatrix} {}^I\boldsymbol{R}_O \boldsymbol{l}_i \\ \boldsymbol{\alpha}_i \end{bmatrix}, \ s \in [t_j, t_j + T_P], \tag{32d}$$

$$\hat{\boldsymbol{v}}_O(s) = \boldsymbol{J}_{i_O} \boldsymbol{J}_i(\hat{\boldsymbol{q}}_i(s)) \dot{\hat{\boldsymbol{q}}}_i(s), \qquad s \in [t_j, t_j + T_P], \tag{32e}$$

$$\hat{\boldsymbol{x}}_i(s) \in X_i, \qquad s \in [t_j, t_j + T_P], \tag{32f}$$

$$\hat{\boldsymbol{\tau}}_i(s) \in T_i, \qquad s \in [t_j, t_j + T_P], \tag{32g}$$

$$\hat{\boldsymbol{x}}(t_j + T_P) \in \mathscr{E}_f \tag{32h}$$

[4] Except from a set of measure zero [23].

where $\mathscr{E}_f$ is a terminal region around the desired trajectory profile that can be appropriately tuned [22]. 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 $E(\cdot) = \hat{\boldsymbol{x}}_O^\top \boldsymbol{P}_x \hat{\boldsymbol{x}}_O + \hat{\boldsymbol{v}}_O^\top \boldsymbol{P}_v \hat{\boldsymbol{v}}_O$, respectively, with $\boldsymbol{P}_x$, $\boldsymbol{P}_v$, $\boldsymbol{Q}_x$, $\boldsymbol{Q}_v$ and $\boldsymbol{R}$ being positive definite matrices to be appropriately tuned [29]. 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 (32b). This means that $\hat{\boldsymbol{x}}_i(s),\ s \in [t_j, t_j+T_P]$ is the solution of (22) based on the measurement of the state at time instance t_j (i.e., $\boldsymbol{x}_i(t_j)$, provided by the onboard navigation system) 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 (32a)-(32h) 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}:

$$\boldsymbol{\tau}_i(s; \boldsymbol{x}(t_j)) = \hat{\boldsymbol{\tau}}_i^*(s; \boldsymbol{x}(t_j)),\ s \in [t_j, t_j+h] \tag{33}$$

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.

Remark 3 (Real world application): We consider that each UVMS is equipped with the appropriate sensor suite in order to measure its position and velocity. In order for an underwater vehicle to estimate its full state vector i.e 3D position, orientation, linear and angular velocities and accelerations it needs to fuse data from the following sensors: a) Attitude and Heading Reference System (AHRS) for the measurement of 3D linear accelerations, angular velocities and angles, b) Doppler Velocity Log (DVL) sensor for the measurement of 3D linear body velocities, Ultra Short Baseline (USBL) for 3D position with respect to an absolute (i.e the acoustic transpoder) frame, d) a pressure based depth sensor. Additionally, the sensor suite can be further enhanced with the appropriate perception sensors such as standard or RGB-D cameras, as well as Imaging Sonars for a 3D representation of the environment. The aforementioned sensors are de facto standard in underwater robotics and when fused with the appropriate estimators (e.g Kalman Fiters, Complementary Filters, Adaptive Monte Carlo Localization Filters) can deliver accurately the full state vector of the UVMS as well as the representation of the operating workspace including the position of obstacles within [30]. Regarding the position of the arm, simple encoders attached to the joints are enough to compute the kinematic chain of the arm. Hence, the complete state estimation of a UVMS in real world applications is feasible with standard sensor and data fusion technologies.
The pseudo-code description of the proposed real-time control scheme for UVMS $i,\ i \in \mathcal{N}$ is given in *Algorithm1*:

Algorithm 1 Real time MPC algorithm:

```
Triggering time   ▷ At time instance $t_j$ UVMS $i$ measures its state vector $\boldsymbol{x}_i$
    $\boldsymbol{p}_i(t_j) \leftarrow$ eq.(2)   ▷ calculates its EE pose
    $\boldsymbol{v}_i(t_j) \leftarrow$ eq.(3)   ▷ calculates its EE velocity
    $\boldsymbol{v}_O(t_j), \boldsymbol{x}_O(t_j) \leftarrow$ eq.$(12)-(13)$   ▷ calculates object pose and velocity
    $\boldsymbol{x}_O^d(s), \boldsymbol{v}_O^d(s),\ s \in [t_j, t_j+T_P] \leftarrow$ eq.$(23), (24)$   ▷ propagating for the time interval $s,\ s \in [t_j, t_j+T_P]$ a map of safe/desired trajectory and velocity of the object
    $\hat{\boldsymbol{\tau}}_i^*(s; \boldsymbol{x}(t_j)),\ s \in [t_j, t_j+T_P] \leftarrow$ FHOCP($\boldsymbol{x}_i(t_j)$)   ▷ Run FHOCP of (32a)-(32h). The solution is an optimal control input trajectory for the time interval $[t_j, t_j+T_P]$.
for $s \in [t_j, t_j+h]$ do
        Apply the $\boldsymbol{\tau}_i(s; \boldsymbol{x}(t_j)) = \hat{\boldsymbol{\tau}}_i^*(s; \boldsymbol{x}(t_j))$ to the UVMS.
    $t_{j+1} = t_j + h$   ▷ The next triggering time
goto Triggering.
```

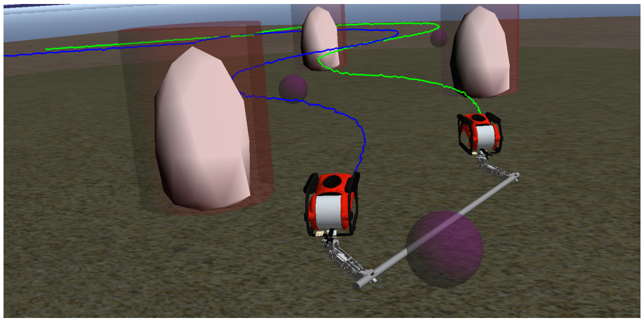


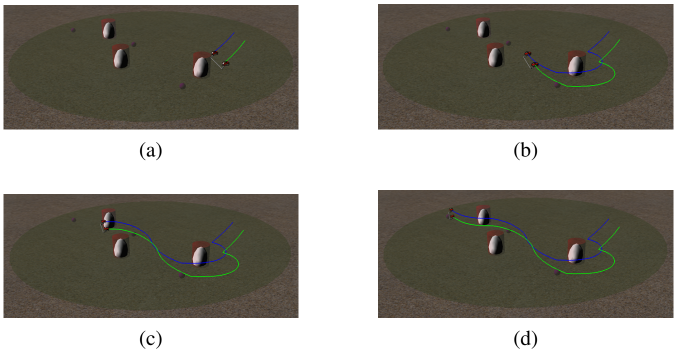
Fig. 2. Simulation environment: Object transportation using two UVMSs inside a constrained workspace including obstacles.

IV. Results

In this section, the effectiveness of the proposed control strategy is verified through both simulation and experimental studies. The simulation results were conducted using a dynamic simulation environment based on the UwSim simulator [31] running on the Robot Operating System (ROS) [32]. The experimental results were conducted in a test tank employing two small UVMSs with in-house built underwater manipulators.

A. Simulation study

We consider a scenario involving 3D motion with two UVMSs with the same structure, transporting a bar-shaped object in a constrained workspace with static obstacles (see Fig.2). The UVMS model is a underwater robotic vehicle equipped with a small 4 DoF manipulator attached at the bow of the vehicle (see Fig.2). The dynamic parameters of the vehicle have been identified via a proper identification scheme [30], 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 [30]. The Constrained NMPC employed in this work is implemented using the NLopt Optimization library [33]. In the following simulation, the objective for the team of UVMSs is to track 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: $\boldsymbol{x}_{obs_1} = [4,\ -4.5]$,


(a)

(b)

(c)

(d)

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

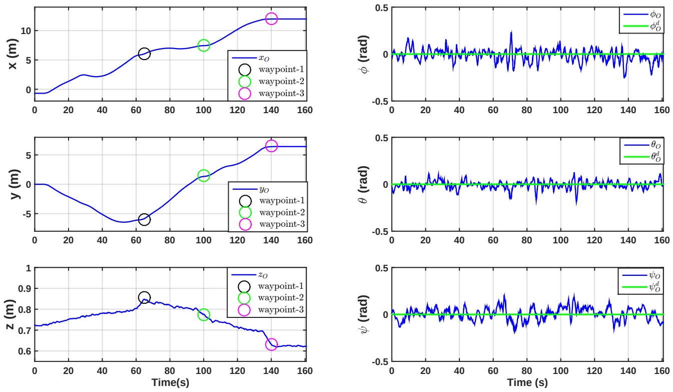


Fig. 4. Simulation study: object coordinates during the control operation

$\boldsymbol{x}_{obs_2} = [9, \ -1.5]$ 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 (23)-(24) was designed with gain $K_{NF} = 0.5$. According to constraints (29), we consider that the vehicle's velocity must not exceed $0.5m/s$ for translation and $0.1rad/s$ for rotational. In the same vein the manipulator joint velocities must be retained between $(-0.1, 0.1)rad/s$. Moreover, the manipulator joint positions (30) must be retained between $(-2,2)rad$. Furthermore, input saturations (31) for the vehicle and manipulator are considered as: $\bar{\tau}_v = 10N$ and $\bar{\tau}_m = 2N$, respectively. The sampling time (25) 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 the 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 $\boldsymbol{x}_{O_1}^d = [6, \ -6, \ 0.85, \ 0, \ 0, \ 0]$, $\boldsymbol{x}_{O_2}^d = [7.5, \ 1.5, \ 0.78, \ 0, \ 0, \ 0]$ and $\boldsymbol{x}_{O_3}^d = [12, \ 6.5, \ 0.65, \ 0, \ 0, \ 0]$ which make the mission more challenging considering the obstacles' positions within the workspace (See Fig.3 and Fig.2). The results are presented in Fig.3-Fig.8. The trajectory of the system within the workspace as well as the 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 avoided the obstacles. The evolution of $det(\boldsymbol{J}(\boldsymbol{q})[\boldsymbol{J}(\boldsymbol{q})]^\top)$ (see (4) and (28)) during the operation is given in Fig.5. It can be easily seen that

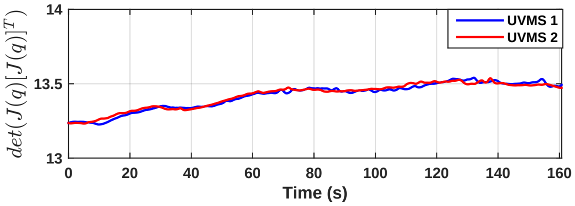


Fig. 5. Simulation study: $det(\boldsymbol{J}(\boldsymbol{q})[\boldsymbol{J}(\boldsymbol{q})]^\top)$ during the control operation

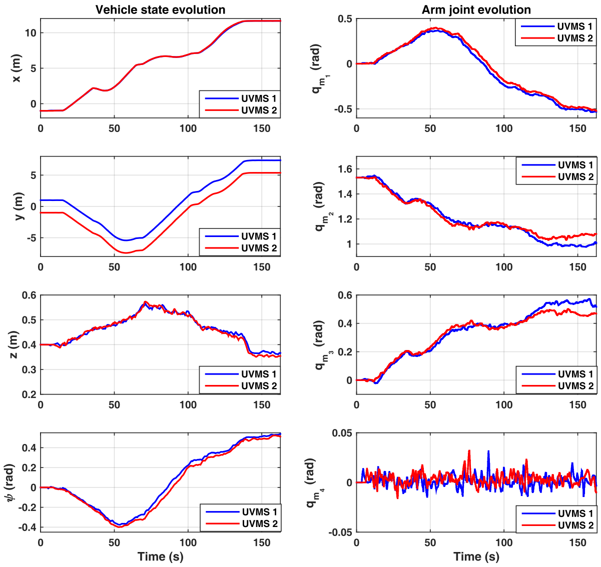


Fig. 6. Simulation study: the evolution of the system states at joint level

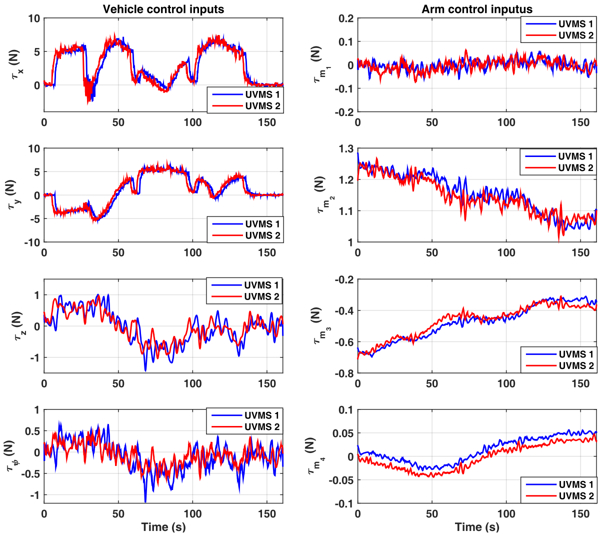


Fig. 7. Simulation study: the control input signals during the control operation

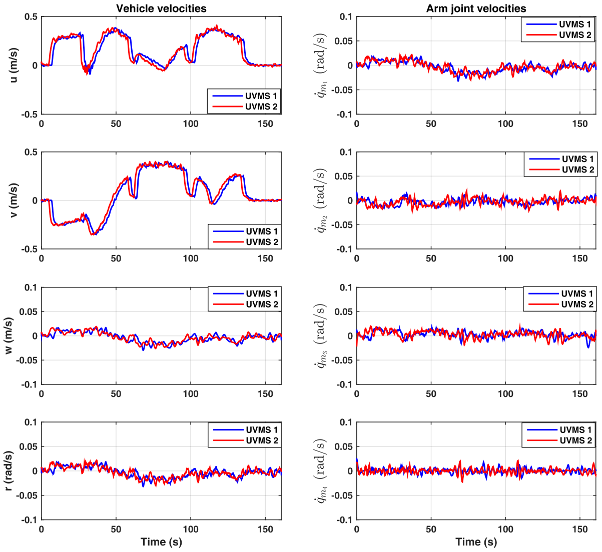


Fig. 8. Simulation study: the evolution of the system velocities at joint level

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.

B. Experimental Study

This section demonstrates the efficacy of the proposed cooperative control scheme via a set of real-time experiments employing two small UVMSs equipped with in-house built underwater manipulators, carrying a common object (see Fig. 9). In particular, Subsection IV-B1 introduces the experimental setup and Subsection IV-B2 presents the detailed results of two cases of experimental studies.

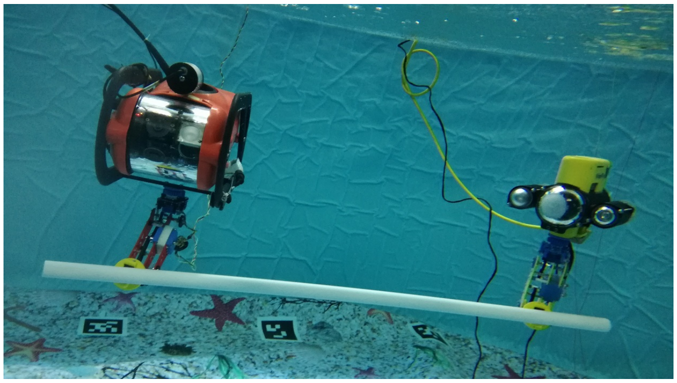


Fig. 9. Experimental setup: small custom made UVMSs under cooperative transportation in real time experimental study. The vehicles used in this work were a 4 DoFs Seabotix LBV and a 3 DoFs VideoRay PRO presented in red and yellow color respectively.

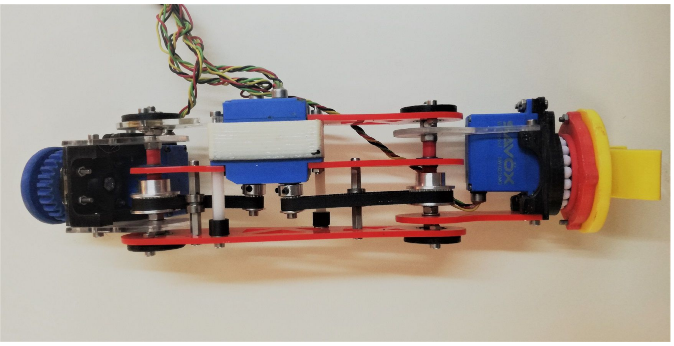


Fig. 10. Custom made 4 Dofs waterproof robotic manipulator used in experimental study.

1) Experimental Setup: The experiments were carried out inside the *NTUA, Control Systems Lab* test tank, with dimensions $5m \times 3m \times 1.5m$ (Fig. 9). The bottom of the tank is covered by a custom-made poster with various visual features and markers. In the following experiments, the team of UVMSs consists of two small ROVs equipped with the same custom made small waterproof manipulator (see Fig.9). More specifically, a 4 DoFs Seabotix LBV (red color), actuated in Surge, Sway, Heave and Yaw and a 3 DoFs VideoRay PRO (yellow color) effective only in Surge, Heave and Yaw motion were used as the vehicle bases in this work (see Fig. 9). Notice that the 3 Dofs VideoRay robot is under-actuated along the Sway axis. This intuitively means that while the combined vehicle-manipulator system is full-actuated at the end-effector frame, the vehicle base remains underactuated along the Sway body frame axis. Thanks to the nature of the optimization procedure, the aforementioned difficulty is handled within the FHOCP (32a)-(32h), which results in a solution that combines the optimal vehicle and manipulator motion in order to achieve the desired movement at the end effector frame. Both of the underwater vehicles are equipped with the same custom made small waterproof manipulator (see Fig. 10). The design parameters of the manipulators are given in Tables I-II. Finally, the object of interest was a pipe with 1.0 m length and 0.35 kg weight. Both UVMSs are equipped with a down-looking Sony PlayStation Eye camera, with 640×480 pixels at 30 frames per second (fps) enclosed in

TABLE I
DENAVIT-HARTENBERG PARAMETERS OF THE ROBOTIC ARM

Link	d_i	θ_i	a_i	α_i
1	L_1	q_1	0	$-\frac{\pi}{2}$
2	0	$q_2 - \frac{\pi}{2}$	L_2	0
3	0	$q_3 + \frac{\pi}{2}$	$-L_3$	$\frac{\pi}{2}$
4	L_4	q_4	0	0

TABLE II
PARAMETERS OF THE ROBOTIC ARM

Parameter	Value	Unit
Link 1 Length(L_1)	0.077	m
Link 2 Length(L_2)	0.147	m
Link 3 Length(L_3)	0.028	m
Link 4 Length(L_4)	0.075	m
Link 1 Mass	0.1	kg
Link 2 Mass	0.2	kg
Link 3 Mass	0.1	kg
Link 4 Mass	0.12	kg
Link Diameter	0.06	m

a waterproof housing. An underwater laser pointer projecting a green dot at the bottom of the test tank is rigidly attached on the vehicle with its axes aligned to the down-looking camera axis. The visual projection of the laser pointer dot on the image plane, along with various data from vehicle's onboard navigation system sensors (e.g., IMU) are used within a proper sensor fusion algorithm, in order to provide the vehicle state vectors. The Seabotix LBV and the VideoRay PRO are also equipped with $SBG\ IG-500A$ AHRS and 10 DOF IMU Sensor respectively, delivering temperature-compensated 3D acceleration, angular velocity and orientation measurements at $100Hz$. The marker localization system is based on the $ArUco$ library [34]. For both UVMSs, the complete state vector of the vehicle ($3D$ position, orientation, velocity) as well as the vehicle's dynamic parameters in the following experimental studies are available via the sensor fusion and state estimation module based on the Complementary Filter notion and a proper identification scheme presented in detail in our previous results [30]. The software implementation of the proposed control scheme was conducted in C++ and Python under the Robot Operating System (ROS) [32]. The Nonlinear Model Predictive Controller employed in this work was implemented using the NLopt Optimization library [33] and was running with 1 ms time step, which is common in a real time operation with underwater robotic systems. It is worth mentioning that recent advances in technology (i.e., the new generation of very powerful CPUs) motivated engineers and scientists to develop faster and more efficient solvers (e.g., ACADO [35]) that therefore allow the reliable implementation of NMPC controllers in real-time fast applications. In this work the overall software was running on two conventional laptops (each UVMS was connected to a separate laptop) with 4 cores, 2.80 GHz CPU and 16 GB of RAM.

2) Experimental Results: In order to prove the efficiency of the proposed controller, two experimental sessions are presented, namely Session A and B. In Session A the objective for the team of UVMSs is to stabilize the object cooperatively in a desired configuration within the test tank while the objective in Session B is to track a set of predefined waypoints while collaboratively carrying the object. Moreover, in the following experiments the team of UVMS should simultaneously avoid the workspace (test tank) boundaries which were modeled according to the spherical world representation. Notice that the small and limited size of the available test tank, did not allow us to consider obstacles inside the workspace. However, collision avoidance with the test tank boundaries, demonstrates the efficacy of the proposed scheme for avoiding collisions in a real time manner. The radius of the sphere $\mathcal{B}(\boldsymbol{p}_i, \bar{r})$ which covers all the UVMS volume (i.e., main body of the vehicle, additional equipment and robotic manipulator for all possible configurations) is defined as $\bar{r} = 0.75m$. Moreover, the radius of the sphere $\mathcal{B}(\boldsymbol{x}_O, r_O)$ that covers the object is defined as $r_O = 0.5m$. Similar to the simulation part, the Navigation function (23)-(24) was designed with gain $K_{NF} = 0.1$. Regarding to constraints (29), in both experiments, we consider that the vehicle's velocity must not exceed $0.5m/s$ for translation and $0.5rad/s$ for rotational velocities. In the same vein, the manipulator joint velocities must be retained between $(-0.2, 0.2)rad/s$. Moreover, the manipulator joint positions (30) must be retained between $(-2.5, 2.5)rad$ for the first joint q_{m_1}, $(-1.5, 0.7)rad$ for the second joint q_{m_2}, and $(-0.5, 1.5)rad$ for the third joint q_{m_3}, respectively. Notice that the fourth joint q_{m_4} is limit free and thus no joint limit constraints considered to this joint. Furthermore, input saturations (31) for the vehicle and manipulator are considered as: $\bar{\tau}_v = 2N$ and $\bar{\tau}_m = 0.2N$, respectively. The sampling time (25) and the prediction horizon for both of the following experiments are $h = 0.15sec$ and $T_p = 5 \times h = 0.75sec$ respectively. Moreover, the matrices $\boldsymbol{P}_x$, $\boldsymbol{Q}_x$, $\boldsymbol{Q}_v$ and $\boldsymbol{R}$ as well as the load sharing coefficients c_1 and c_2 for both UVMSs are equal and set to: $\boldsymbol{P}_x = \boldsymbol{Q}_x = 0.5 \cdot \boldsymbol{I}_{6\times6}$, $\boldsymbol{R} = 0.15 \cdot \boldsymbol{I}_{8\times8}$, $\boldsymbol{Q}_v = 0.2 \cdot \boldsymbol{I}_{6\times6}$, and $c_1 = c_2 = 0.5$.

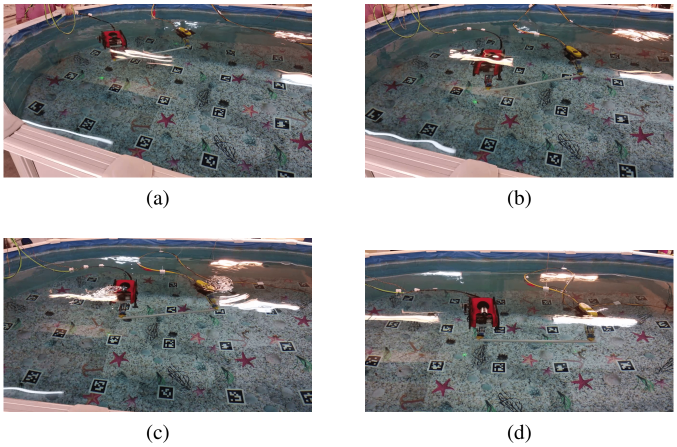

(a)
(b)
(c)
(d)

Fig. 11. Experimental study - stabilization scenario: The evolution of the proposed methodology in 4 consecutive time instants.

Session A: Stabilization

As stated before, the objective for the team of UVMS in the first experiment is to transfer and stabilize the object into a desired configuration inside the small test tank. More specifically, the desired configuration was set as $\boldsymbol{x}_O^d = [0.0,\ 0.0,\ 0.4,\ 0,\ 0.45,\ 1.57]$. The results are presented in Fig.11-Fig.15. The trajectory of the system within the workspace as well as the object coordinates evolution are depicted in Fig. 11 and Fig.12 respectively. It can be seen that the team of UVMSs have successfully transported and stabilized the object to the desired configuration. The evolution of the system velocity at joints level as well as the corresponding control inputs are indicated in Fig.13 and Fig.14, respectively. Moreover, the evolution of the system's velocities is given in Fig.15. It can be seen easily that all of the aforementioned values remained in their corresponding constraints sets during the experiment's evolution.

Session B: Waypoint tracking

In the following experiment, the objective for the team of UVMSs is to track a set of predefined waypoints within the workspace while collaboratively carrying the object. More specifically, 4 waypoints were set as given in Table-III, where

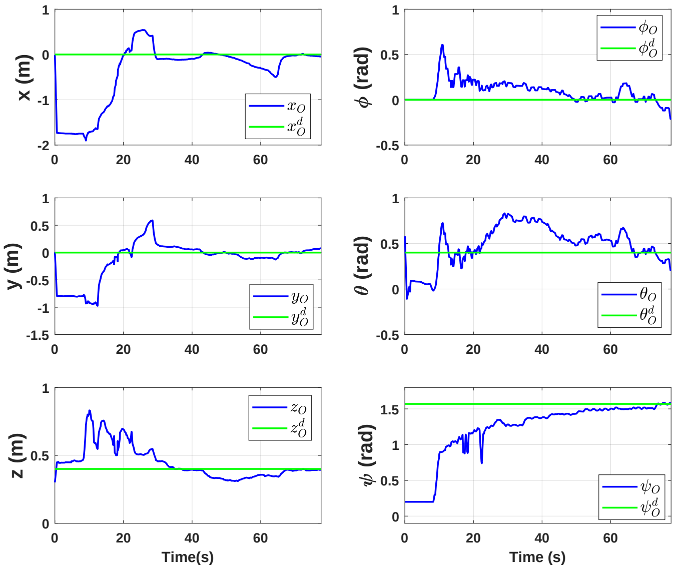


Fig. 12. Experimental study - stabilization scenario: Object coordinates during the control operation

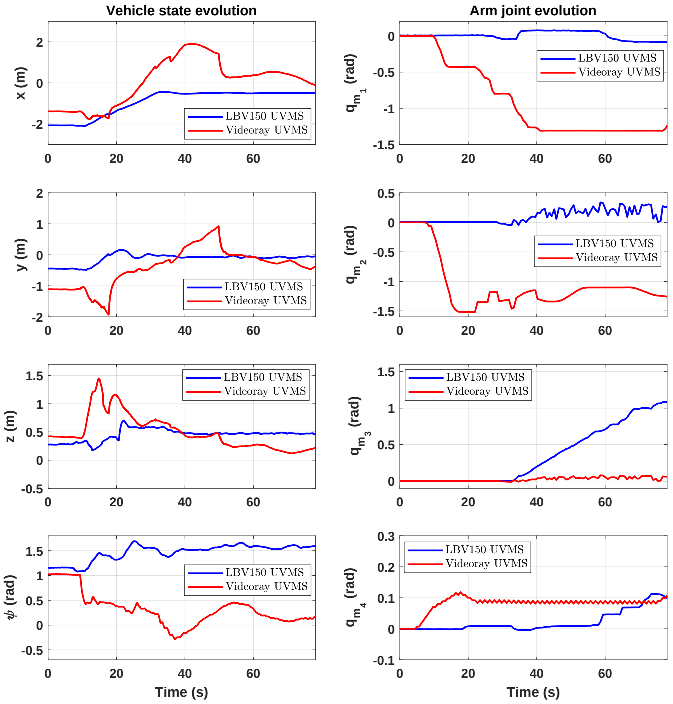


Fig. 13. Experimental study - stabilization scenario: The evolution of the system states at joint level

a successive setpoint changes where the object position is lower than a user-designed threshold with respect to its desired waypoint configuration. Please notice that the mission here is more challenging regarding the stabilization scenario and demonstrates the repeatability of the proposed control strategy in successive experiment in our small water tank. The results are presented in Fig.16-Fig.19. The object coordinates evolution is depicted in Fig. 17. It can be seen that the UVMSs have successfully transported cooperatively the object and have tracked the set of predefined way points while safely avoided

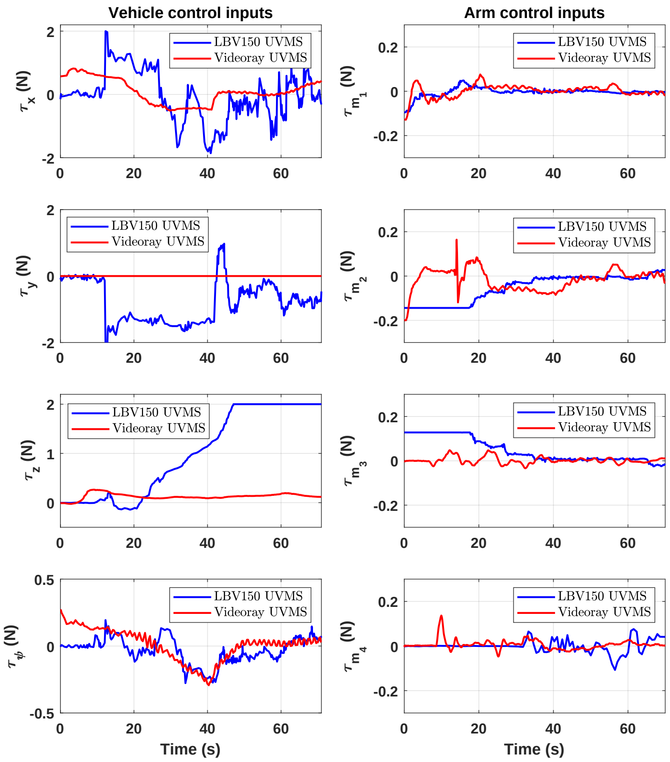


Fig. 14. Experimental study - stabilization scenario: The control input signals during the control operation

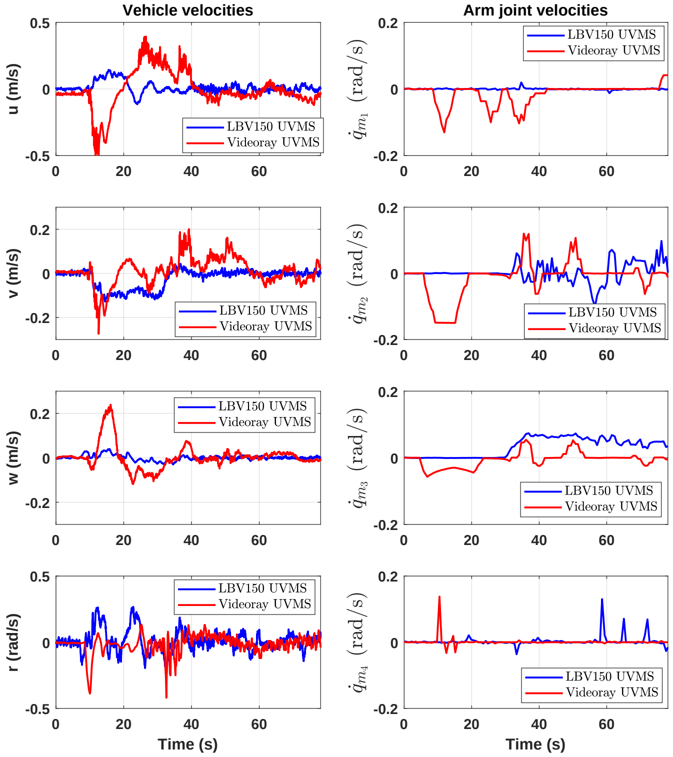


Fig. 15. Experimental study- stabilization scenario: The evolution of the system velocities at joint level

TABLE III
THE SET OF PREDEFINED WAYPOINTS

Waypoint	Predefined values for each elements					
	x_O^d	y_O^d	z_O^d	ϕ_O^d	θ_O^d	ψ_O^d
$\boldsymbol{x}_{O_1}^d$	0.0	0.0	0.4	0.0	0.45	1.5
$\boldsymbol{x}_{O_2}^d$	1	0.5	0.4	0.0	0.45	0.0
$\boldsymbol{x}_{O_3}^d$	-0.7	-0.5	0.4	0.0	0.45	0.75
$\boldsymbol{x}_{O_4}^d$	0.0	0.0	0.4	0.0	0.45	1.5

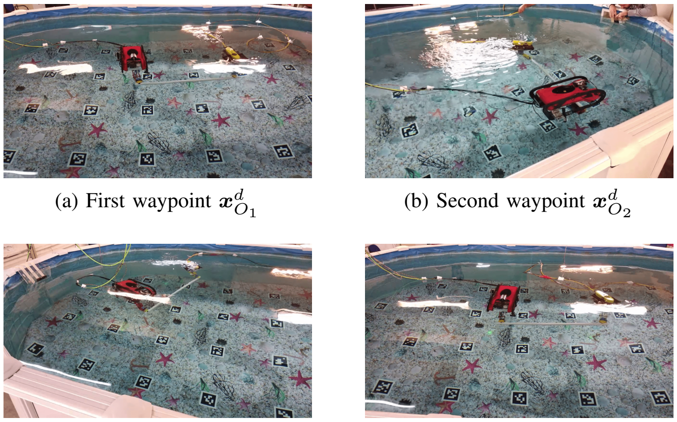

(a) First waypoint $\boldsymbol{x}_{O_1}^d$
(b) Second waypoint $\boldsymbol{x}_{O_2}^d$
(c) Third waypoint $\boldsymbol{x}_{O_3}^d$
(d) Forth waypoint $\boldsymbol{x}_{O_4}^d$

Fig. 16. Experimental study - Waypoint tracking scenario: The evolution of the proposed methodology in 4 consecutive time instants.

any collision with the test tank boundaries. The evolution of the system states and velocities at joints level as well as the corresponding control inputs are indicated in Fig.18 Fig.19 and Fig.20, respectively. As it was expected from the theoretical findings, in both of the considered scenarios, all of the system values were retained in the corresponding constraints sets defined by the respective bounds and consequently all of the system constraints were satisfied. Two videos demonstrating the aforementioned realistic simulation and the experimental results of the proposed methodology can be found in at the urls: i) Simulation: https://youtu.be/O5WHov7EGMI and ii) Experiment: https://youtu.be/eIu57Ftxb1g.

V. Summary and Future Work

In this work, we presented a novel object transportation control scheme for a team UVMSs operating in a constrained workspace with static obstacles. Various limits 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 hence reducing the need for high bandwidth requirements during explicit information exchange (e.g via acoustic modems). Moreover, the control scheme adopts load sharing among the UVMSs according to their specific payload capabilities. Future research efforts will be devoted towards in investigatigation collaborative grasping and transportation strategies for objects

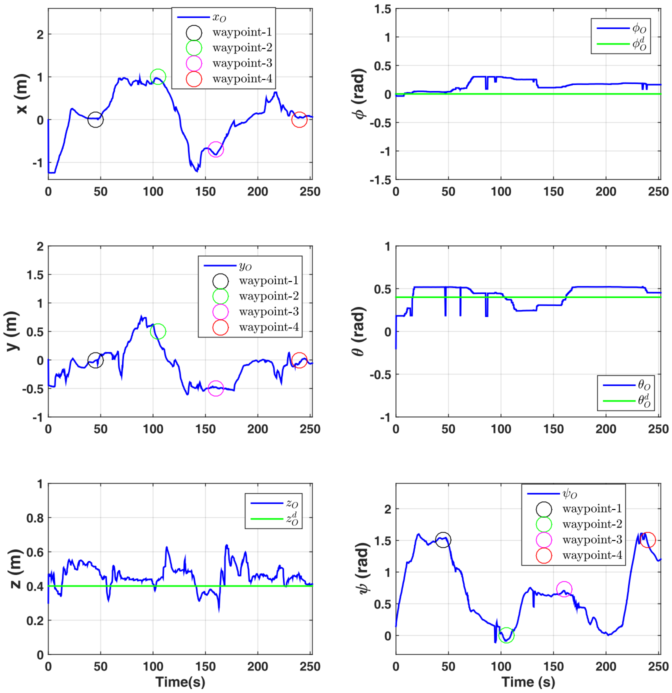


Fig. 17. Experimental study - Waypoint tracking scenario: Object coordinates during the control operation

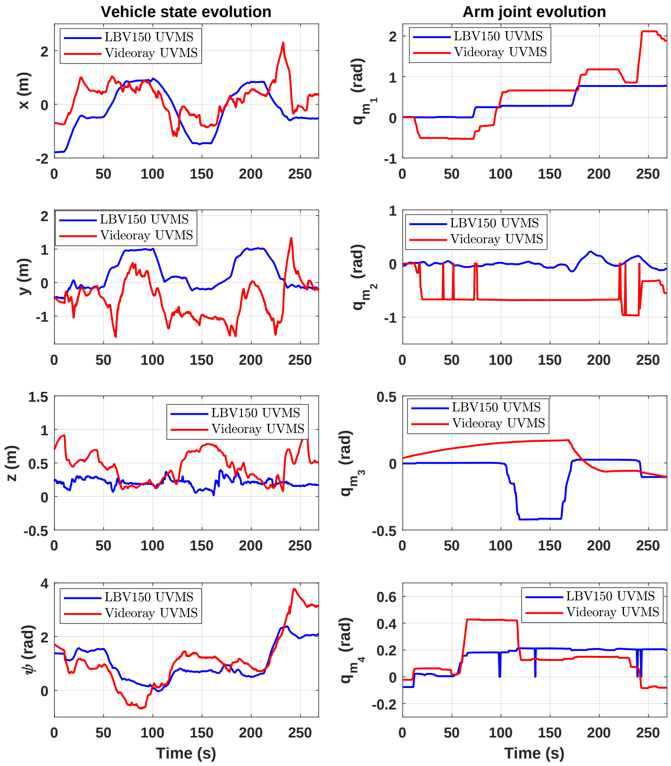


Fig. 18. Experimental study - Waypoint tracking scenario: The evolution of the system states at joint level

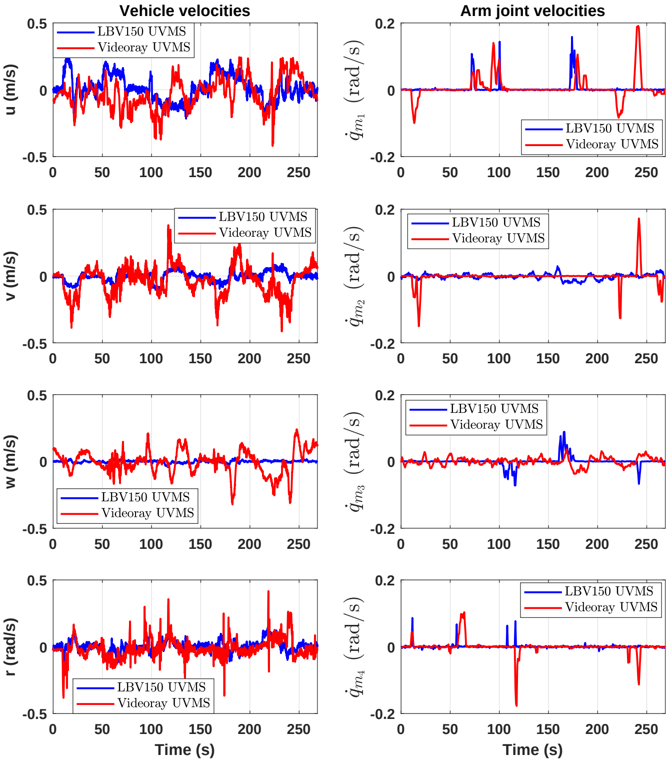


Fig. 19. Experimental study - Waypoint tracking scenario: The evolution of the system velocities at joint level

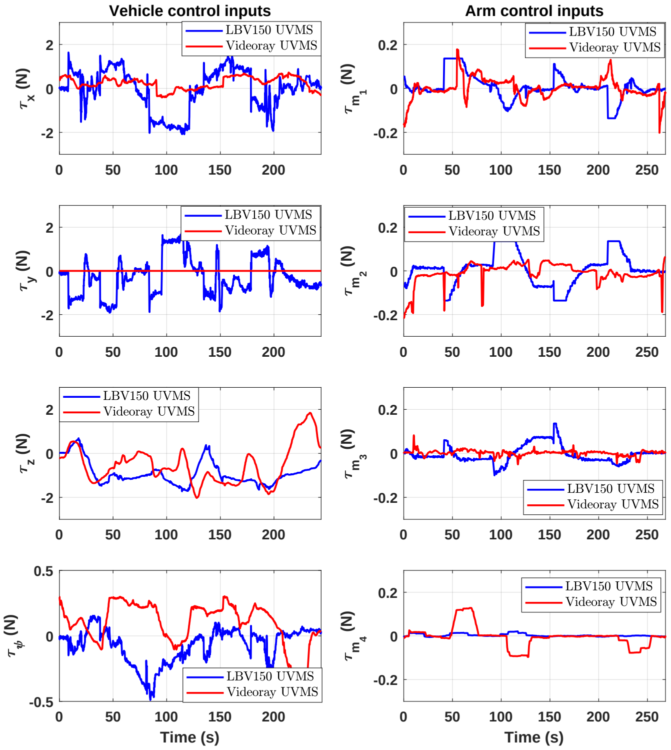


Fig. 20. Experimental study - Waypoint tracking scenario: The control input signals during the control operation

with increased geometric complexity, identifying the optimal regions for safe grasp (e.g no slippage) while taking into account how these regions will affect the manipulability of the augmented (UVMSs and object) system during transportation tasks.

References

[1] T. Fossen, "Guidance and control of ocean vehicles," *Wiley, New York*, 1994.

[2] 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.

[3] 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.

[4] P. Londhe, S. Mohan, B. Patre, and L. Waghmare, "Robust task-space control of an autonomous underwater vehicle-manipulator system by pid-like fuzzy control scheme with disturbance estimator," *Ocean Engineering*, vol. 139, pp. 1–13, 2017.

[5] E. Simetti and G. Casalino, "Whole body control of a dual arm underwater vehicle manipulator system," *Annual Reviews in Control*, vol. 40, pp. 191–200, 2015.

[6] D. Lane, D. O'Brien, M. Pickett, J. Davies, G. Robinson, D. Jones, E. Scott, G. Casalino, G. Bartolini, G. Cannata, A. Ferrara, D. Angelleti, M. Coccoli, G. Veruggio, R. Bono, P. Virgili, M. Canals, R. Pallas, E. Gracia, and C. Smith, "Amadeus: Advanced manipulation for deep underwater sampling," *IEEE Robotics and Automation Magazine*, vol. 4, no. 4, pp. 34–45, 1997.

[7] V. Rigaud, . Coste-Manire, M. Aldon, P. Probert, M. Perrier, P. Rives, D. Simon, D. Lane, J. Kiener, A. Casals, J. Amat, P. Dauchez, and M. Chantler, "Union: Underwater intelligent operation and navigation," *IEEE Robotics and Automation Magazine*, vol. 5, no. 1, pp. 25–34, 1998.

[8] E. Simetti, G. Casalino, S. Torelli, A. Sperind, 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.

[9] D. M. Lane, F. Maurelli, P. Kormushev, M. Carreras, M. Fox, and K. Kyriakopoulos, "Persistent autonomy: the challenges of the pandora project," *IFAC Proceedings Volumes*, vol. 45, no. 27, pp. 268–273, 2012.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] S. Heshmati-Alamdari, G. Karras, and K. Kyriakopoulos, "A distributed predictive control approach for cooperative manipulation of multiple underwater vehicle manipulator systems," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 4626–4632, 2019.

[21] F. Allgwer, 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.

[22] S. Heshmati-Alamdari, A. Nikou, and D. V. Dimarogonas, "Robust trajectory tracking control for underactuated autonomous underwater vehicles in uncertain environments," *IEEE Transactions on Automation Science and Engineering*, pp. 1–14, 2020.

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

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

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

[26] S. Soylu, B. Buckham, and R. Podhorodeski, "Redundancy resolution for underwater mobile manipulators," *Ocean Engineering*, vol. 37, no. 2-3, pp. 325–343, 2010.

[27] 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.

[28] 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.

[29] S. Heshmati-Alamdari, G. Karras, P. Marantos, and K. Kyriakopoulos, "A robust model predictive control approach for autonomous underwater vehicles operating in a constrained workspace," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 6183–6188, 2018.

[30] G. C. Karras, C. P. Bechlioulis, P. Marantos, S. Heshmati-alamdari, and K. J. Kyriakopoulos, "Sensor-based motion control of autonomous underwater vehicles, part i: modeling and low-complexity state estimation," *Autonomous Underwater Vehicles: Design and Practice*, pp. 15–43, 2020.

[31] 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.

[32] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, (Kobe, Japan), May 2009.

[33] S. G. Johnson, "The nlopt nonlinear-optimization package." http://ab-initio.mit.edu/nlopt. [Online; accessed 27-May-2021],(2021).

[34] S. Garrido-Jurado, R. M. noz Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280 – 2292, 2014.

[35] B. Houska, H. Ferreau, and M. Diehl, "An Auto-Generated Real-Time Iteration Algorithm for Nonlinear MPC in the Microsecond Range," *Automatica*, vol. 47, no. 10, pp. 2279–2285, 2011.

Dr. Shahab Heshmati alamdari received a Diploma in Mechanical Engineering, a Masters Degree in Robotics and Automatic Control and a PhD in Mechanical Engineering from the National Technical University of Athens (NTUA), Greece, in 2009, 2012, and 2018, respectively. He has held postdoctoral positions at KTH Royal Institute of Technology, Sweden and TUM Technical University of Munich, Germany. He is currently an Assistant Professor at the Section of Automation & Control, Department of Electronic Systems, Aalborg University, Denmark. His research interests include navigation and sensor based control of unmanned autonomous vehicles, the task planning and distributed control of multi-robot systems, such as free flying manipulators and autonomous robotic vehicles.

Dr. George C. Karras is an Assistant Professor at the University of Thessaly, Greece, since 2018 and a Research Scientist at the Control Systems Lab, National Technical University of Athens (NTUA), since 2012. He received a Diploma in Mechanical Engineering in 2003 (NTUA), a Masters Degree in Robotics and Automatic Control in 2006 (NTUA) and a PhD in Mechanical Engineering in 2011 (NTUA). He has worked on many R&D projects in the field of Robotics and Embedded Control Systems both in Academic Institutes and Industry. His research includes sensor-based control of unmanned robotic vehicles (underwater, ground and aerial), system identification, navigation and visual servo control.

Prof. Kostas J. Kyriakopoulos received a Diploma in Mechanical Engineering (Honors) from NTUA, in 1985 and the MS & Ph.D. in Electrical, Computer & Systems Engineering (ECSE) from Rensselaer Polytechnic Institute (RPI), Troy, NY in 1987 and 1991, respectively. From 1988 to 1991 he did research at the NASA Center for Intelligent Robotic Systems for Space Exploration. Between 1991-93 he was an Assistant Professor at ECSE-RPI and the NY State Center for Advanced Technology in Automation and Robotics. Since 1994 he has been with the Control Systems Laboratory of the Mechanical Engineering Department at NTUA, where he currently serves as a Professor and Director. His current interests are in the area of Nonlinear Control Systems applications in Sensor Based Motion Planning & Control of multi-Robotic Systems: Manipulators & Vehicles (Mobile, Underwater and Aerial) and Micro- & Bio- Mechatronics and he has contributed to a large number of projects funded by the European Commission and the Greek secretariat for Research & Technology. He holds the title of IEEE Fellow (RAS).