Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-31 (2022) 110–117

Kinematic and Dynamic Control of Cooperating Underwater Vehicle-Manipulator Systems ⋆

Markus H. Iversflaten * **Aurora Haraldsen** *
Kristin Y. Pettersen *

* *Centre for Autonomous Marine Operations and Systems (AMOS), Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway (e-mail: {markus.h.iversflaten, aurora.haraldsen, kristin.y.pettersen}@ntnu.no).*

Abstract: Cooperative control of autonomous underwater robots can provide capabilities that exceed those of any single marine robot. In this paper, the cooperation between two different underwater vehicle-manipulator systems is considered. Cooperation is enabled through a control scheme which generates state-dependent reference signals to achieve synchronization of the end-effector positions of the robots. These references are tracked by the individual control systems, which consist of decoupled kinematic and dynamic controllers. Consequently, the control systems do not constrain the choice of cooperation scheme. Due to the kinematic redundancy of the robots, a task-priority inverse kinematics controller is used for velocity-level redundancy resolution, and several tasks are defined for each robot. Furthermore, the dynamics of the robots are controlled by a higher-order sliding mode controller, namely the generalized super-twisting algorithm with adaptive gains. The dynamic control approach provides robustness to model uncertainties and unknown disturbances. The effectiveness of the methods is verified in a simulation study on a heterogeneous pair of UVMSs.

Keywords: Cooperative control, underwater vehicle-manipulator systems, sliding mode control, inverse kinematics control, autonomous underwater vehicles

1. INTRODUCTION

Cooperation between heterogeneous teams of underwater vehicle-manipulator systems (UVMSs) for inspection and intervention tasks can provide capabilities that exceed those of any existing marine robots. The need for subsea inspection, maintenance and repair (IMR) operations is strongly increasing as subsea structures, particularly from the oil and gas industry, are ageing, and offshore wind-farms and aquaculture are emerging. Subsea operations are dangerous, expensive, time-consuming, and increasingly rely on marine robots that can alleviate the challenges of the harsh subsea environment. The underwater remotely operated vehicle (ROV) is well established as an efficient robotic tool in such operations. Even so, no marine robot possesses all the capabilities that are necessary in subsea IMR. Rather than creating a robot that is a jack of all trades, it may be more tractable to rely on teams of existing marine robots that complement each other. Despite the potential advantages of employing teams of different robots, such approaches are barely represented in literature.

Existing marine robots include work-class ROVs, observation ROVs, survey autonomous underwater vehicles (AUVs), and articulated intervention AUVs (AIAUVs). The typical work-class ROV is tethered, has hovering capabilities, and has manipulator arms that can perform heavy lifting. The major drawbacks of these vehicles are their big size and weight, which demand the deployment of a support vessel and a crew in order to operate. Consequently, the costs associated with work-class ROVs are typically high, in addition to their significant carbon footprint. Moreover, their large size restricts access. Observation ROVs are smaller vehicles that have improved access capabilities and cost less to operate, but lack the heavy-lifting capabilities that work-class ROVs have. They are also tethered. Survey AUVs are untethered vehicles, which are normally torpedo shaped in order to minimize the hydrodynamic drag forces. Their capabilities are limited to survey missions due to their lack of hovering capabilities and manipulator arms. The AIAUV is a slender, untethered vehicle with hovering capabilities. It is in itself a manipulator arm, as its body is composed of links and joints. However, it is quite lightweight and therefore lacks the pure strength of a work-class ROV. Marine robots that can interact with their environment, such as work-class ROVs and AIAUVs, are classified as UVMSs. In this paper, this class of vehicles is studied.

⋆ This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, through the ERC Advanced Grant 101017697-CRÈME. The work is also supported by the Research Council of Norway through the Centres of Excellence funding scheme, project No. 223254 - NTNU AMOS, and by VISTA - a basic research program in collaboration between The Norwegian Academy of Science and Letters, and Equinor.

10.1016/j.ifacol.2022.10.417

Cooperative control for free-floating robots is highly complex, and only a few studies on the topic exist. In Padir and Koivo (2003), kinematic and dynamic models for two cooperating UVMSs transporting an object are developed. However, the UVMSs are identical, and control design is only briefly discussed. The focus is shifted towards kinematic control and redundancy resolution in Padir (2005), involving the same vehicles. Cooperative transportation is also presented in Simetti and Casalino (2017). Special emphasis is here put on the development of a unifying architecture for the kinematic control of both individually and cooperatively operating UVMSs. The dynamics of the systems are controlled with a proportional-integral (PI) controller. The Italian research project MARIS (Casalino et al., 2016) has explored cooperation of floating manipulators as one of its main goals. In Simetti et al. (2009), it was shown that strict cooperative tasks for mobile manipulators can be traced back to the same conceptual and algorithmic task-priority framework developed for individual mobile manipulators. Later works by Simetti are extended to the case of UVMSs, such as Simetti et al. (2015), where a cooperative task priority control policy between two UVMSs was developed. A common factor in the previously mentioned works is the use of homogeneous UVMSs during cooperation. To our knowledge, there exist little to no works that have studied the topic of cooperation between heterogeneous UVMSs.

In this paper, the cooperative control problem of a heterogeneous team of two UVMSs is addressed, both in terms of kinematic and dynamic control. Specifically, we consider the cooperation between a work-class UVMS and an AIAUV. The work-class UVMS, exemplified by Phoenix (Healey and Lienard, 1993), has a large and heavy base, with a manipulator arm mounted on its underside. The AIAUV, exemplified by Eelume (Schmidt-Didlaukies et al., 2018), is multi-articulated, light, and slender. They both possess more than six degrees of freedom (DOF), i.e. they are redundant with respect to their end-effector configuration. To exploit these redundancies, we apply a task-priority inverse kinematics controller. In a task-priority framework, it is possible to define several tasks that should be completed beyond the main control objective. However, the dynamics of UVMSs are generally very hard to model. The complexity can mainly be attributed to the many hydrodynamic effects that are experienced underwater. Furthermore, underwater vehicles are subject to external disturbances. The dynamic control of the UVMS therefore requires robust methods, which we choose to achieve using a sliding mode controller. Specifically, we apply the generalized super-twisting algorithm with adaptive gains (AGSTA) proposed by Borlaug et al. (2020). The control approach is identical for both UVMSs, which is a testament to its robustness. The resulting system consists of the two UVMSs controlled independently, i.e. in a distributed manner, on both the kinematic and dynamic levels. Cooperation between the UVMSs is enabled purely through reference signal generation. The cooperative control scheme can therefore easily be replaced or extended. Moreover, the chosen method tightly synchronizes the positions of the UVMS end-effectors, while their orientation may be chosen freely.

The paper is organized as follows. The mathematical model of a general UVMS is presented in Section 2. The cooperative control scheme is described in Section 3. The kinematic and dynamic controllers are presented in Sections 4 and 5, respectively. In Section 6, the proposed methods are validated through a simulation study. Lastly, Section 7 presents conclusions and future work.

2. UNDERWATER VEHICLE-MANIPULATOR SYSTEMS

Both cooperative agents considered in this study belong to the class of UVMSs. They are characterized by having a floating base attached to, or in some cases consisting of, a robot manipulator. UVMSs typically have propulsion systems that are able to displace the entire vehicle, as well as separate actuators that can control the manipulator joints individually. In this section, we present the kinematic and dynamic modeling of such systems.

We consider the dynamics of the system expressed in a body-fixed reference frame, i.e. a frame that is rigidly attached to, and moving along with, the base of the vehicle. The equations of motion for a general UVMS may be written as (From et al., 2014)

$$\dot{\boldsymbol{\xi}} = \boldsymbol{J}_{\boldsymbol{\xi}}(\boldsymbol{q})\boldsymbol{\zeta}, \quad (1)$$

$$\boldsymbol{M}(\boldsymbol{\theta})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}(\boldsymbol{\theta},\boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{\theta},\boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{g}(\boldsymbol{\xi}) = \boldsymbol{\tau}. \quad (2)$$

The system configuration of a UVMS with n joints is here defined as $\boldsymbol{\xi} \triangleq \left[\boldsymbol{\eta}^T, \boldsymbol{\theta}^T\right]^T$, where $\boldsymbol{\eta} \triangleq \left[(\boldsymbol{p}_{nb}^n)^T, \boldsymbol{q}^T\right]^T \in \mathbb{R}^7$ represents the position and attitude of the base frame with respect to the inertial frame and $\boldsymbol{\theta} \in \mathbb{R}^n$ is the vector of joint angles. A unit quaternion parametrization of attitude is employed, defined as $\boldsymbol{q} \triangleq \left[\eta, \boldsymbol{\epsilon}^T\right]^T \in \mathbb{R}^4$, where $\eta \in \mathbb{R}$ is the real part and $\boldsymbol{\epsilon} \in \mathbb{R}^3$ is the vector part of the quaternion. The vector of body-fixed velocities of a UVMS is denoted as $\boldsymbol{\zeta} \triangleq \left[\boldsymbol{v}^T, \boldsymbol{\omega}^T, \dot{\boldsymbol{\theta}}^T\right]^T \in \mathbb{R}^{6+n}$, where $\boldsymbol{v} \triangleq [u, v, w]^T$ are the linear velocities, $\boldsymbol{\omega} \triangleq [p, q, r]^T$ are the angular velocities, and $\dot{\boldsymbol{\theta}} \triangleq [\theta_1, \ldots, \theta_n]^T$ are the n joint velocities. The Jacobian matrix $\boldsymbol{J}_{\boldsymbol{\xi}}(\boldsymbol{q}) \triangleq \text{diag}\{\boldsymbol{R}_{nb}(\boldsymbol{q}), \boldsymbol{T}_{\boldsymbol{q}}(\boldsymbol{q}), \boldsymbol{I}_n\}$ relates the body-fixed velocities to the inertial frame by the relationship (1). The inertia and added inertia of the system are collected in the matrix $\boldsymbol{M}(\boldsymbol{\theta})$. The matrix $\boldsymbol{C}(\boldsymbol{\theta},\boldsymbol{\zeta})$ contains Coriolis and centripetal terms. Hydrodynamic damping terms are collected in the matrix $\boldsymbol{D}(\boldsymbol{\theta},\boldsymbol{\zeta})$. The restoring forces due to gravity and buoyancy effects are given in $\boldsymbol{g}(\boldsymbol{\xi})$. Lastly, $\boldsymbol{\tau}$ is the vector of control forces and moments acting on the system. The modeling of these terms can be found in Antonelli (2018), along with the complete expression for $\boldsymbol{J}_{\boldsymbol{\xi}}(\boldsymbol{q})$.

3. COOPERATIVE CONTROL SCHEME

The main objective in the cooperation between the UVMSs is the coordination of the end-effector positions and orientations. This is e.g. a necessary part in achieving joint transportation of a rigid object, carried out by two UVMSs. Firstly, their end-effectors must be positioned correctly relative to each other and the object at all times. Secondly, the orientation of their end-effectors should be chosen so that they can handle the object most efficiently.

The latter choice will depend on the mechanical compositions of the manipulator arms.

To achieve such coordination, we use a method inspired by Mora-Aguilar et al. (2018), wherein a similar objective was defined for a pair of mobile robot manipulators. The basic idea of the method is to employ an operation point and a virtual line that passes through the operation point. The operation point is the mean end-effector position of the two UVMSs, and the virtual line is a vector between the end-effectors of the UVMSs. In this way, a trajectory can be defined in which the robots move together in a coordinated fashion, with respect to the operation point but also with respect to each other. In this method, the robots are coupled at a kinematic level. This is beneficial during cooperation, as it forces the robots to move together and also react to deviations on an individual level. For example, a perturbation acting on only the AIAUV will result in increased errors in both robots' objectives and they will resolve this jointly, rather than individually. The operation point and virtual line are denoted $\boldsymbol{P}$ and $\boldsymbol{L}$ respectively and are given by

$$\boldsymbol{P} = \frac{1}{2}\left(\boldsymbol{p}_e^P + \boldsymbol{p}_e^E\right), \tag{3}$$

$$\boldsymbol{L} = \boldsymbol{p}_e^P - \boldsymbol{p}_e^E, \tag{4}$$

where $\boldsymbol{p}_e^P, \boldsymbol{p}_e^E$ are the end-effector positions of the work-class UVMS and the AIAUV given in the inertial frame, respectively. Contrary to Mora-Aguilar et al. (2018), both of the Jacobians are well-defined on $\mathbb{R}^3$ and given by

$$\boldsymbol{J}_{\boldsymbol{P}} = \frac{1}{2}\begin{bmatrix}\boldsymbol{I}_3 & \boldsymbol{I}_3\end{bmatrix}, \tag{5}$$

$$\boldsymbol{J}_{\boldsymbol{L}} = \begin{bmatrix}\boldsymbol{I}_3 & -\boldsymbol{I}_3\end{bmatrix}. \tag{6}$$

We may now describe the kinematic relationship as

$$\dot{\boldsymbol{\rho}} = \begin{bmatrix}\boldsymbol{J}_{\boldsymbol{P}} \\ \boldsymbol{J}_{\boldsymbol{L}}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{p}}^P \\ \dot{\boldsymbol{p}}^E\end{bmatrix} \coloneqq \boldsymbol{J}_{PL}\dot{\boldsymbol{p}}_e^{PE}, \tag{7}$$

where $\boldsymbol{\rho} = \left[\boldsymbol{P}^T, \boldsymbol{L}^T\right]^T$. The relationship between the linear end-effector and body-fixed velocities are given by

$$\dot{\boldsymbol{p}}_e^{PE} = \underbrace{\begin{bmatrix}\boldsymbol{R}_{be}^P(\boldsymbol{q}) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{R}_{be}^E(\boldsymbol{q})\end{bmatrix}}_{\boldsymbol{J}_{pos}}\boldsymbol{v}_b^{PE}, \tag{8}$$

where $\boldsymbol{v}_b^{PE} = \left[\left(\boldsymbol{v}_b^P\right)^T, \left(\boldsymbol{v}_b^E\right)^T\right]^T$ and $\boldsymbol{R}_{be}(\boldsymbol{q})$ is the rotation matrix relating the linear velocity of the end-effector frame to the linear velocity of the base frame. Inserting (8) into (7) yields the complete kinematic model relating the virtual line and operating point to the body-fixed linear velocities:

$$\dot{\boldsymbol{\rho}} = \boldsymbol{J}_{PL}\boldsymbol{J}_{pos}\boldsymbol{v}_b^{PE}. \tag{9}$$

The reference signal $\boldsymbol{v}_b^{PE}$ is then tracked through the highest prioritized task of the kinematic controller proposed in Section 4.

Note that the orientation of the end-effectors may be chosen independently of $\boldsymbol{P}$ and $\boldsymbol{L}$. Normally, it should be chosen based on the physical parameters of the manipulator arms as well as the control objective. During transit, the robots may assume their most energy efficient configuration. On the other hand, the robots may maximize lifting capabilities during a joint transportation mission. In the present work, different setpoints in orientation are given in order to demonstrate the potential of the method.

As the control structure described in this paper is completely decentralized and decoupled, cooperation schemes are easily interchangeable. This enables efficient transitioning between different subsea tasks.

4. KINEMATIC CONTROL

In this section we present the kinematic control methods of the UVMSs, which are necessary for making the UVMSs follow the references produced by the cooperative control scheme. The kinematic control module generates velocity references for the dynamic controller proposed in Section 5. We choose to employ a task-priority control scheme, as both vehicles are kinematically redundant, and may thus solve tasks beyond the immediate coordination of end-effectors.

4.1 Inverse Kinematics

In order to find suitable system trajectories to complete the control objectives, we need to solve the inverse kinematics problem. Let $\boldsymbol{x} \in \mathbb{R}^m$ denote a task of dimension m, and let the relation between the task variable and the system configuration $\boldsymbol{\xi}$ be given by

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{\xi}). \tag{10}$$

The above kinematic relation must be inverted and solved for $\boldsymbol{\xi}$. However, since (10) is nonlinear in general, it is difficult to obtain a trajectory $\boldsymbol{\xi}_d$ which corresponds to the desired task trajectory $\boldsymbol{x}_d$. Time differentiation of (10) and insertion of (1) yields the first-order differential kinematics

$$\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{f}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\dot{\boldsymbol{\xi}} = \bar{\boldsymbol{J}}(\boldsymbol{\xi})\dot{\boldsymbol{\xi}} = \boldsymbol{J}(\boldsymbol{\xi})\boldsymbol{\zeta}, \tag{11}$$

where $\boldsymbol{J}(\boldsymbol{\xi}) \triangleq \bar{\boldsymbol{J}}(\boldsymbol{\xi})\boldsymbol{J}_{\boldsymbol{\xi}}(\boldsymbol{q}) \in \mathbb{R}^{m\times(6+n)}$ maps the body-fixed velocities to the m-dimensional task-space velocities. This is a linear relationship, and hence, it is desirable to solve the inverse kinematics problem on a differential level.

4.2 Singularity-Robust Task-Priority Redundancy Resolution

In a task-priority framework, it is possible to design and complete several tasks at once. This corresponds to solving (11) for several different functions $\boldsymbol{f}_i, i \in 1, \ldots, k$, where k is the number of tasks.

Consider an optimization with two tasks, where the first task has priority over the other. We may solve (11) with respect to task $\boldsymbol{x}_a$, whilst minimizing the error of the second task $\boldsymbol{x}_b$ as (Chiaverini, 1997)

$$\boldsymbol{\zeta}_d = \boldsymbol{J}_a^{\dagger}\dot{\boldsymbol{x}}_{a,d} + \boldsymbol{N}_a\boldsymbol{J}_b^{\dagger}\dot{\boldsymbol{x}}_{b,d}. \tag{12}$$

Since the task Jacobians generally are not square, the right Moore-Penrose pseudoinverse is applied, i.e. $\boldsymbol{J}^{\dagger} \triangleq \boldsymbol{J}^T\left(\boldsymbol{J}\boldsymbol{J}^T\right)^{-1}$. Moreover, the null space projector $\boldsymbol{N}_a$ projects the velocity reference $\boldsymbol{J}_b^{\dagger}\dot{\boldsymbol{x}}_{b,d}$ into the null space of $\boldsymbol{J}_a$ by $\boldsymbol{N}_a(\boldsymbol{\xi}) \triangleq \left(\boldsymbol{I} - \boldsymbol{J}_a^{\dagger}(\boldsymbol{\xi})\boldsymbol{J}_a(\boldsymbol{\xi})\right)$. In practice, this allows us to choose the task $\boldsymbol{x}_b$ arbitrarily and it will not interfere with the completion of task $\boldsymbol{x}_a$. However, completion of task $\boldsymbol{x}_b$ or any lower level tasks is not guaranteed. A generalization of (12) to an arbitrary number of tasks can be found in Antonelli (2009). Since (12) is prone to drifting

when integrated to obtain $\boldsymbol{\xi}$, we introduce feedback into the control law (12) as (Chiacchio et al., 1991)

$$\boldsymbol{\zeta}_d = \boldsymbol{J}_a^{\dagger}(\dot{\boldsymbol{x}}_{a,d} + \boldsymbol{K}_a\tilde{\boldsymbol{x}}_a) + \boldsymbol{N}_a\boldsymbol{J}_b^{\dagger}(\dot{\boldsymbol{x}}_{b,d} + \boldsymbol{K}_b\tilde{\boldsymbol{x}}_b), \quad (13)$$

where the task-coordinate errors are defined as $\tilde{\boldsymbol{x}} \triangleq \boldsymbol{x}_d - \boldsymbol{x}$ and $\boldsymbol{K}_a, \boldsymbol{K}_b > 0$ are gain matrices.

4.3 Tasks

The choice of tasks in a task-priority framework varies greatly depending on the application. In this work, the main objective is to synchronize the end-effectors of the two UVMSs. The other tasks are designed to aid the completion of the main objective. In the case that all defined tasks are compatible, they will all converge asymptotically (Borlaug et al., 2020). Task compatibility occurs when two tasks use the same DOF of a system that is redundant with respect to this DOF. Typically, this results in successive convergence of tasks, starting with the highest prioritized task. Moreover, if two arbitrary tasks a and b are independent, such that we have $\text{rank}\big(\boldsymbol{J}_a^T\big) + \text{rank}\big(\boldsymbol{J}_b^T\big) = \text{rank}\big(\big[\boldsymbol{J}_a^T\ \boldsymbol{J}_b^T\big]\big)$, then they will converge simultaneously (Antonelli, 2009).

The implemented tasks are the same for the work-class UVMS and the AIAUV, and we let the highest prioritized task be the end-effector configuration. This task is directly related to the cooperation scheme described by (9). The position and orientation of the end-effector are obtained through forward kinematics, and the orientation of the end-effector is represented by the unit quaternion. The orientation error of the base frame is given through the quaternion product $\tilde{\boldsymbol{q}} \triangleq \boldsymbol{q}_d \otimes \boldsymbol{q}^*$, where $\boldsymbol{q}_d$ is the desired orientation and $\boldsymbol{q}^* \triangleq [\eta, -\boldsymbol{\epsilon}^T]^T$ is the conjugate of the quaternion. The quaternion product is defined as

$$\begin{aligned} \tilde{\eta} &\triangleq \eta_d\eta + \boldsymbol{\epsilon}_d^T\epsilon, \\ \tilde{\boldsymbol{\epsilon}} &\triangleq \eta\boldsymbol{\epsilon}_d - \eta_d\boldsymbol{\epsilon} + S(\boldsymbol{\epsilon})\boldsymbol{\epsilon}_d. \end{aligned} \quad (14)$$

The quaternion error of two aligned frames is $\tilde{\boldsymbol{q}} = \big[1, \boldsymbol{0}^T\big]^T$. It is sufficient to represent this error as the three-dimensional vector part $\tilde{\boldsymbol{\epsilon}}$, as $\tilde{\eta}$ is given through the unit quaternion constraint $\eta^2 + \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} = 1$. The orientation of the end-effector will be denoted by $\boldsymbol{q}_e \triangleq \big[\eta_e, \boldsymbol{\epsilon}_e^T\big]^T$, and the position of the end-effector will be denoted by $\boldsymbol{p}_{ne}^n$. It follows that the end-effector configuration task is defined by

$$\boldsymbol{x}_a \triangleq \begin{bmatrix} \boldsymbol{p}_{ne}^n \\ \boldsymbol{\epsilon}_e \end{bmatrix} \in \mathbb{R}^6, \quad (15)$$

$$\dot{\boldsymbol{x}}_a \triangleq \boldsymbol{J}_a(\boldsymbol{\xi})\boldsymbol{\zeta}, \quad (16)$$

where the Jacobian is given by

$$\begin{aligned} \boldsymbol{J}_a(\boldsymbol{\xi}) &\triangleq \boldsymbol{J}_{\boldsymbol{\xi}}(\boldsymbol{q})\boldsymbol{J}_e(\boldsymbol{\theta}) \\ &= \begin{bmatrix} \boldsymbol{R}_{nb}(\boldsymbol{q}) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{T}_{\boldsymbol{\epsilon}}(\boldsymbol{q}) \end{bmatrix} \boldsymbol{J}_e(\boldsymbol{\theta}) \in \mathbb{R}^{6\times(6+n)}, \end{aligned} \quad (17)$$

$$\boldsymbol{T}_{\boldsymbol{\epsilon}}(\boldsymbol{q}) \triangleq \frac{1}{2}\left(\eta\boldsymbol{I}_3 + S(\boldsymbol{\epsilon})\right). \quad (18)$$

The end-effector Jacobian $\boldsymbol{J}_e(\boldsymbol{\theta})$ relates the end-effector velocities to the base frame and is given by forward kinematics of the manipulator, see e.g. Antonelli (2018). The term $S(\boldsymbol{\epsilon}) \in \text{so}(3)$ represents the skew-symmetric form of $\boldsymbol{\epsilon}$.

We let the second most prioritized task be the base attitude. Let $\boldsymbol{\Theta}_{nb} \triangleq [\phi, \theta, \psi]^T$ be the Euler angle representation of the attitude, denoting the roll, pitch, and yaw angles, repectively. This task will be defined slightly differently for the two vehicles. Thus, the super- and subscripts "P" and "E" are used to denote the work-class UVMS and the AIAUV in the task definitions, respectively. For UVMSs such as Phoenix, it is desireable to keep the base level with respect to the horizontal plane. We therefore choose the base attitude task for the work-class UVMS as the minimization of its roll and pitch angles ϕ and θ. As . The task is thus given by

$$\boldsymbol{x}_b^P \triangleq [\phi_P\ \theta_P]^T \in \mathbb{R}^2, \quad (19)$$

$$\dot{\boldsymbol{x}}_b^P \triangleq \boldsymbol{J}_{b,P}(\boldsymbol{\xi}_P)\boldsymbol{\zeta}_P, \quad (20)$$

where the Jacobian is

$$\boldsymbol{J}_{b,P}(\boldsymbol{\xi}_P) \triangleq \begin{bmatrix} \boldsymbol{0}_{2\times3} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{T}_{nb}\big(\boldsymbol{\Theta}_{nb}^P\big) & \boldsymbol{0}_{2\times n}, \end{bmatrix}. \quad (21)$$

The Jacobian $\boldsymbol{J}_{b,P}(\boldsymbol{\xi}_P) \in \mathbb{R}^{2\times(6+n)}$ and transformation matrix $\boldsymbol{T}_{nb}(\boldsymbol{\Theta}_{nb})$ relates the body-fixed angular velocity vector $\boldsymbol{\omega}$ to the Euler rate vector $\dot{\boldsymbol{\Theta}}_{nb}$. The AIAUV considered in this work is passively stable in roll. Hence, we choose to only stabilize the pitch of the AIAUV:

$$x_b^E \triangleq \theta_E \in \mathbb{R}, \quad (22)$$

$$\dot{x}_b^E \triangleq \boldsymbol{J}_{b,E}(\boldsymbol{\xi}_E)\boldsymbol{\zeta}_E. \quad (23)$$

The Jacobian is then given by

$$\boldsymbol{J}_{b,E}(\boldsymbol{\xi}_E) \triangleq \big[\boldsymbol{0}_{1\times3}\ [0\ 1\ 0]\,\boldsymbol{T}_{nb}\big(\boldsymbol{\Theta}_{nb}^E\big)\ \boldsymbol{0}_{1\times n}\big] \in \mathbb{R}^{6+n}. \quad (24)$$

Note that $\boldsymbol{T}_{nb}(\boldsymbol{\Theta}_{nb})$ is singular for $\theta = \pm\frac{\pi}{2}$. However, this is not regarded as an issue since the task should prevent the pitch of the UVMSs from approaching these angles.

In order to keep the manipulator joints within their mechanical limits, we define a task that limits their movement to a defined range. The task does not enter as a third task but may rather be seen as an implicit task within the task-priority framework. We define the task by choosing a weight matrix by the method in Sarkar et al. (1999) and use this matrix to take the weighted pseudoinverse of the task Jacobians. The control law in (13) then takes the form

$$\begin{aligned} \boldsymbol{\zeta}_d = &\boldsymbol{J}_{\boldsymbol{W},a}^{\dagger}(\dot{\boldsymbol{x}}_{a,d} + \boldsymbol{K}_a\tilde{\boldsymbol{x}}_a) \\ &+ \boldsymbol{N}_{\boldsymbol{W},a}\boldsymbol{J}_{\boldsymbol{W},b}^{\dagger}(\dot{\boldsymbol{x}}_{b,d} + \boldsymbol{K}_b\tilde{\boldsymbol{x}}_b), \end{aligned} \quad (25)$$

where $\boldsymbol{J}_{\boldsymbol{W}}^{\dagger} \triangleq \boldsymbol{W}^{-1}\boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J})^{-1}$ is the weighted Moore-Penrose pseudoinverse and $\boldsymbol{N}_{\boldsymbol{W}} \triangleq \left(\boldsymbol{I} - \boldsymbol{J}_{\boldsymbol{W}}^{\dagger}\boldsymbol{J}\right)$. Moreover, the weight matrix $\boldsymbol{W}$ is chosen as in Sarkar et al. (1999).

5. DYNAMIC CONTROL

This section describes the dynamic controller which takes the references produced by the kinematic controller and outputs commanded thruster forces and joint torques. In fixed-base robotic systems, it is common to neglect the dynamics based on the assumption that the dynamic loop is much faster than the kinematic loop. However, the dynamics of underwater vehicles are in general slow. Moreover, UVMSs will experience significant coupling forces caused by joint motions (Borlaug et al., 2021), as well as ocean current disturbances and modeling errors due to the complex hydrodynamic forces and moments acting

on them. Therefore, we consider an advanced dynamic controller in order to make the UVMSs precisely track the given reference signals.

5.1 Sliding Mode Control

Due to the challenge of modeling UVMSs and the presence of unknown disturbances, it is necessary to apply a robust control technique. We let the dynamic controller be a sliding mode controller (SMC), due to its insensitivity to uncertain plant parameters and disturbances (Yoerger and Slotine, 1985). Specifically, we choose to use a second-order SMC, which greatly attenuates chattering in the control input; a well-known drawback of conventional sliding mode control (Shtessel et al., 2014). The goal of a SMC is to drive the system dynamics to a surface in the state space $\mathcal{S} = \{x : \sigma(x) = 0\}$, where x is the state and $\sigma(x)$ is the *sliding surface* that is designed such that the error dynamics of the system converge to zero while on $\mathcal{S}$.

5.2 Adaptive Generalized Super-Twisting Algorithm

The super-twisting algorithm (STA) is a powerful second-order SMC (Levant and Levantovsky, 1993). The adaptive generalized super-twisting algorithm (AGSTA) is an SMC inspired by the adaptive STA in Shtessel et al. (2010) and the generalized STA in Castillo et al. (2017). AGSTA works for a class of systems whose perturbations and uncertain control coefficients are time- and state-dependent. The dynamically adapted control gains ensure global finite-time convergence. The control law in Castillo et al. (2017) can be written as

$$\begin{aligned} u_{\text{AGSTA}} &= -k_1\phi_1(\sigma) + z \in \mathbb{R}, \\ \dot{z} &= -k_2\phi_2(\sigma), \end{aligned} \tag{26}$$

where σ is the sliding surface and

$$\phi_1(\sigma) = \lceil\sigma\rfloor^{\frac{1}{2}} + \beta\sigma \tag{27}$$

$$\phi_2(\sigma) = \frac{1}{2}\lceil\sigma\rfloor^0 + \frac{3}{2}\beta\lceil\sigma\rfloor^{\frac{1}{2}} + \beta^2\sigma, \tag{28}$$

with $\lceil a\rfloor^b = |a|^b \operatorname{sgn}(a)$ and $k_1, k_2, \beta > 0 \in \mathbb{R}$ as controller gains. The extra linear term $\beta\sigma$ in (27) helps counteract the effects of state-dependent perturbations, which can increase exponentially in time. We let k_1 and k_2 be adaptive gains defined by the update law (Shtessel et al., 2010)

$$\dot{k}_1 = \begin{cases} \omega_1\sqrt{\dfrac{\gamma_1}{2}}, & \text{if } \sigma \neq 0 \\ 0, & \text{if } \sigma = 0 \end{cases} \tag{29a}$$

$$k_2 = 2\epsilon k_1 + \lambda + 4\epsilon^2, \tag{29b}$$

where $\epsilon, \lambda, \gamma_1, \omega_1 \in \mathbb{R}$ are positive constants.

In the control of the UVMSs, the sliding surface must be chosen such that $\boldsymbol{\sigma} = \mathbf{0} \implies \tilde{\boldsymbol{\xi}} \to \mathbf{0}$. In other words, we want to make $\left(\tilde{\boldsymbol{\xi}}, \tilde{\boldsymbol{\zeta}}\right) = (\mathbf{0}, \mathbf{0})$, where

$$\tilde{\boldsymbol{\xi}} = \begin{bmatrix} \tilde{\boldsymbol{p}}^n_{nb} \\ \tilde{\boldsymbol{\epsilon}} \\ \tilde{\boldsymbol{\theta}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}^n_{nb} - \boldsymbol{p}^n_{nb,d} \\ \eta\boldsymbol{\epsilon}_d - \eta_d\boldsymbol{\epsilon} + \boldsymbol{\epsilon}_d \times \boldsymbol{\epsilon} \\ \boldsymbol{\theta} - \boldsymbol{\theta}_d \end{bmatrix} \tag{30}$$

and $\tilde{\boldsymbol{\zeta}} = \boldsymbol{\zeta} - \boldsymbol{\zeta}_d$, an asymptotically stable equilibrium point of (1)-(2). The velocity reference $\boldsymbol{\zeta}_d$ is given by (25), and $\boldsymbol{\xi}_d$ is found from $\boldsymbol{\zeta}_d$ through integration of (1). We

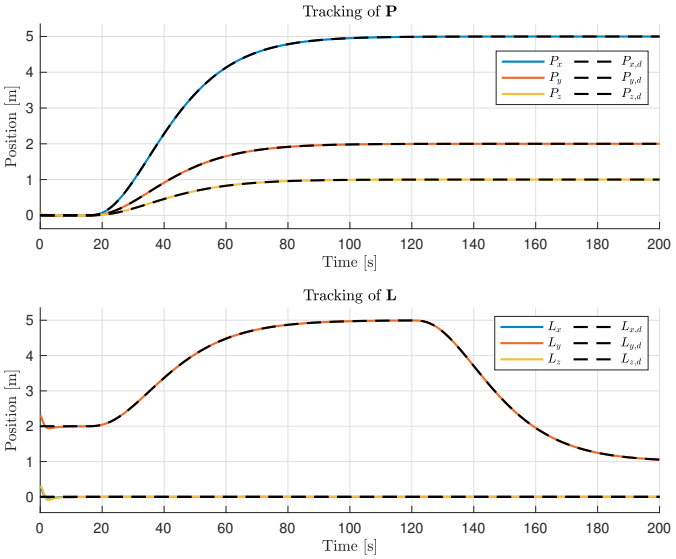

Fig. 1. Tracking of the desired operation point $\boldsymbol{P}_d$ and virtual line $\boldsymbol{L}_d$ (note that L_x and L_z are coinciding).

define the virtual reference vector $\boldsymbol{\zeta}_r = \boldsymbol{\zeta}_d - \boldsymbol{\Lambda}\tilde{\boldsymbol{\xi}}$, with $\boldsymbol{\Lambda} = \operatorname{diag}([\boldsymbol{K}_p\boldsymbol{R}_{nb}(\boldsymbol{q}) \operatorname{sgn}(\tilde{\eta})\boldsymbol{I}_3 \ \boldsymbol{K}_{\boldsymbol{\theta}}])$, where $\boldsymbol{K}_p$ and $\boldsymbol{K}_{\boldsymbol{\theta}}$ are constant, positive definite gain matrices. Let the sliding surface be defined as

$$\boldsymbol{\sigma} = \boldsymbol{\zeta} - \boldsymbol{\zeta}_r \in \mathbb{R}^{6+n}, \tag{31}$$

and let the control input be given by

$$\boldsymbol{\tau}(\boldsymbol{\theta}) = \boldsymbol{u}_{\text{AGSTA}} \in \mathbb{R}^{6+n}, \tag{32}$$

where $\boldsymbol{u}_{\text{AGSTA}}$ is given by (26)-(28). It follows from Theorem 2 in Borlaug et al. (2020) that the sliding surface $\boldsymbol{\sigma} = \mathbf{0}$ is a global finite-time stable equilibrium point of the system (1)-(2). Once the system is confined the sliding manifold, the tracking error dynamics are given by

$$\tilde{\boldsymbol{\zeta}} = -\boldsymbol{\Lambda}\tilde{\boldsymbol{\xi}} \quad \Leftrightarrow \quad \begin{aligned} \tilde{\boldsymbol{v}} &= -\boldsymbol{K}_p\boldsymbol{R}_{nb}(\boldsymbol{q})\tilde{\boldsymbol{p}}^n_{nb} && \text{(33a)} \\ \tilde{\boldsymbol{\omega}} &= -\operatorname{sgn}(\tilde{\eta})\tilde{\boldsymbol{\epsilon}} && \text{(33b)} \\ \dot{\tilde{\boldsymbol{\theta}}} &= -\boldsymbol{K}_{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}} && \text{(33c)} \end{aligned}$$

It is shown in Borlaug et al. (2020) that the tracking errors (33) converge asymptotically to zero. Hence, the UVMSs will track the reference velocities provided by the kinematic control loop, which ultimately leads to the desired coordination of the UVMS end-effectors.

6. SIMULATIONS

In this section, the proposed control methods described in the previous sections are verified through a simulation study. The methods have been implemented on a heterogeneous team of UVMSs, namely the work-class UVMS Phoenix and the AIAUV Eelume.

6.1 Robot Setup

Phoenix is a fully actuated work-class UVMS with a 3-DOF manipulator arm on its underside (Healey and Lienard, 1993; Fossen, 1994; Antonelli, 2018). Its robotic arm is composed of $n_{\mathcal{P}} = 3$ y-revolute joints. Consequently, only the roll and yaw angles of the end-effector are uniquely given by the base. Its thrust configuration matrix $\boldsymbol{B}_P$ is

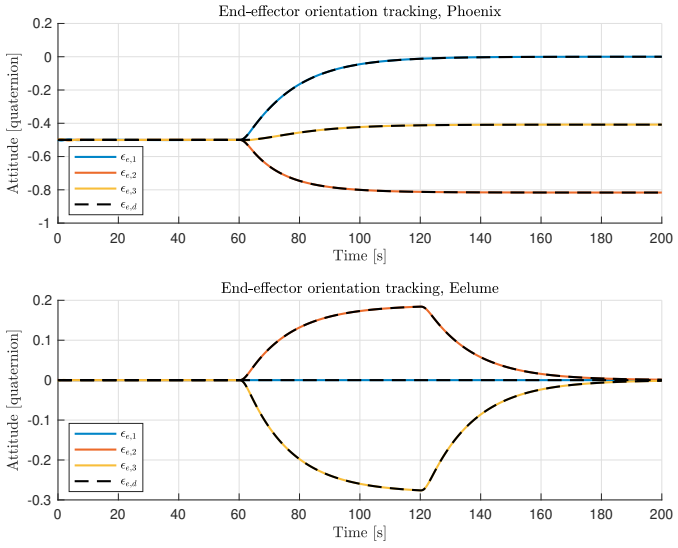

Fig. 2. Desired end-effector orientation tracking, given by the vector part of the quaternion.

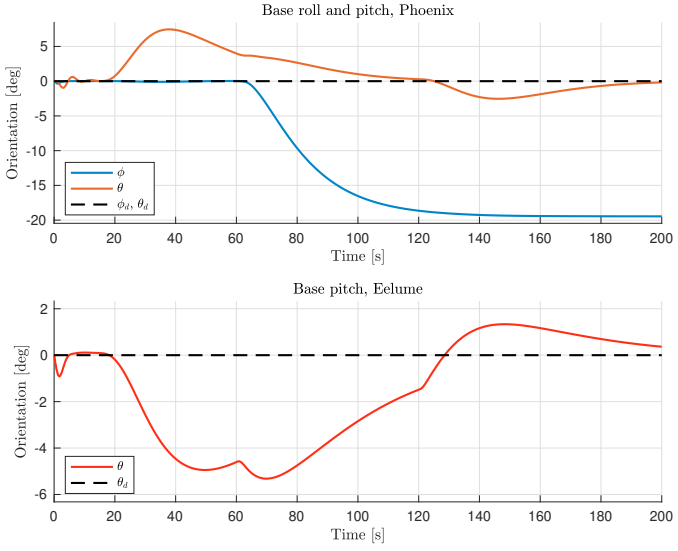

Fig. 3. Base orientation task trajectories; roll and pitch for Phoenix and pitch for Eelume.

here taken to be the identity matrix, i.e. $\boldsymbol{B}_P = \boldsymbol{I}_{6+n_P}$. In total, Phoenix has a mass of 5454kg.

Eelume is an AIAUV (Liljebäck and Mills, 2017; Schmidt-Didlaukies et al., 2018). It is a modular robot; its links and joints may be interconnected in a number of ways. In the present work, Eelume consists of $n_E = 4$ single-axis joints and five links. Two joints are y-revolute while the other two are z-revolute, in order to allow both pitch and yaw motions. The joints can produce a maximum torque of 17Nm each. Despite not having a separate vehicle base, Eelume belongs to the class of UVMSs. For modeling purposes, we consider its first link as its base. Moreover, it has six thrusters that enables propulsion in all DOF. The thrusters are able to produce up to 60N of force. These are distributed along its links, such that its thrust configuration matrix $\boldsymbol{B}_E(\boldsymbol{\theta}_E)$ depends on the joint configuration. Its mass is around 57kg, i.e. just over a hundredth of Phoenix' mass.

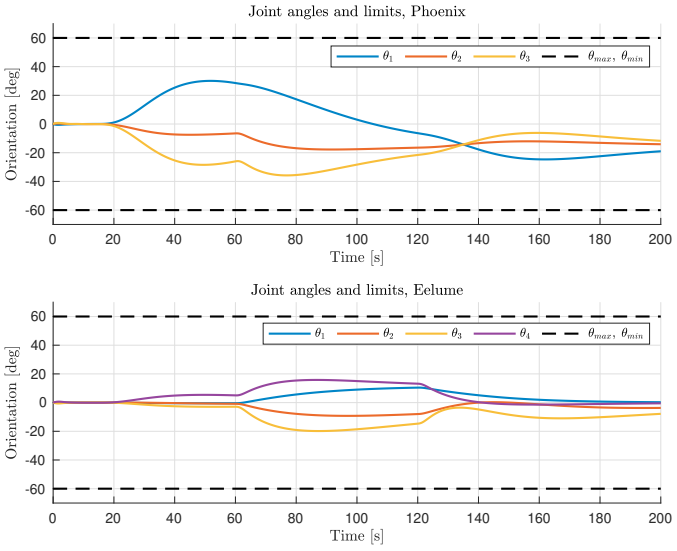

Fig. 4. Joint angle trajectories and joint limits in degrees.

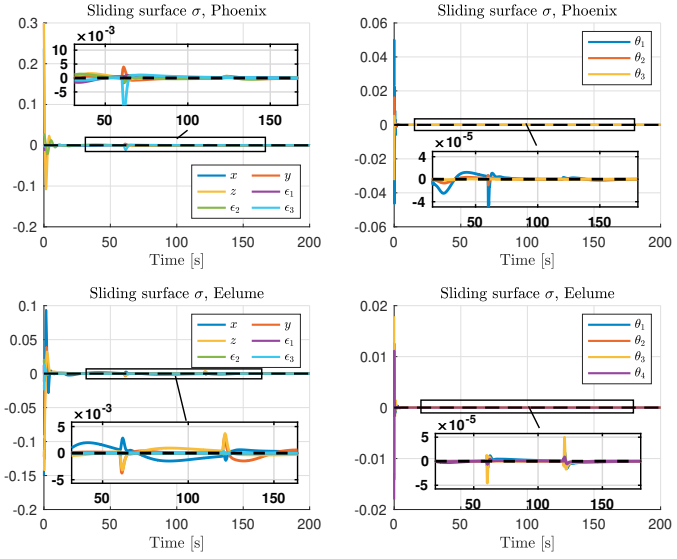

Fig. 5. Sliding surfaces $\boldsymbol{\sigma}_P$ and $\boldsymbol{\sigma}_E$. Left column shows base configuration and right column shows joint angles. Zoom is applied to show details after initial transient.

6.2 Reference Generation

As described in Section 3, cooperation between the UVMSs is enabled by generating state-dependent reference trajectories that are to be tracked cooperatively. Both the desired operation point $\boldsymbol{P}_d$ and the virtual line $\boldsymbol{L}_d$ are given as set-point values. This is also the case for the desired end-effector orientations. The initial desired operation point is chosen as $(0, 0, 0)$m. At $t = 15$s, the desired operation point is changed to $(5, 2, 1)$m and the desired virtual line length $\|\boldsymbol{L}_d\|$ is increased from 2m to 5m. The end-effector orientations of the robots are tracked individually. Each end-effector starts with a constant desired orientation. At $t = 60$s, the desired orientation is changed by giving a step in the desired end-effector quaternion $\boldsymbol{q}_{e,d}$. Later, at $t = 120$s, another reference step is given in which the desired virtual line length is decreased to 1m along with a change in the desired end-effector orientation of Eelume. It is important that all configuration and velocity

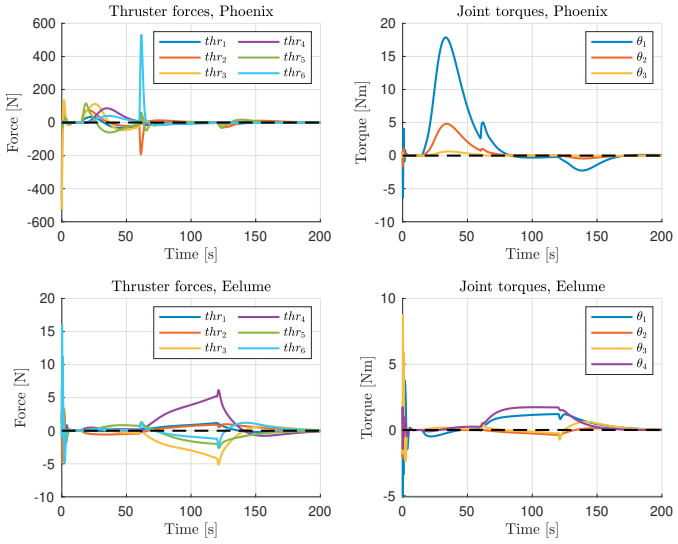

Fig. 6. Applied thruster forces and joint torques of Phoenix and Eelume

references are smooth, due to the adaptive gains (29) in the dynamic controller. Any non-smooth reference signal will cause a non-zero sliding surface (31) and yield increased gains. We therefore choose to use third-order reference models to smoothen the reference signals at the kinematic level. Specifically, a position reference model for $\boldsymbol{\rho}_d$ (recall that $\boldsymbol{\rho}_d = \left[\boldsymbol{P}_d^T, \boldsymbol{L}_d^T\right]^T$) and a quaternion reference model for $\boldsymbol{q}_{e,d}$ are used to smoothen the signals.

6.3 Simulation Parameters

The simulation was run in Matlab with ODE1 and a fixed step size of 0.001s. The kinematic controller gains for Phoenix and Eelume were chosen as

$$\begin{aligned}\boldsymbol{K}_{a,P} &= \text{diag}\left\{0.5, 0.5, 0.5, 0.2, 0.2, 0.2\right\},\\ \boldsymbol{K}_{b,P} &= 3 \cdot \boldsymbol{I}_2,\\ \boldsymbol{K}_{a,E} &= \text{diag}\left\{0.5, 0.5, 0.5, 0.2, 0.2, 0.2\right\},\\ \boldsymbol{K}_{b,E} &= 0.3,\end{aligned}$$

where a and b specify the different tasks which were defined in Section 4.3, and the kinematic control laws are given by (25). Notably, the task gain $\boldsymbol{K}_{b,P}$ is chosen significantly higher than $\boldsymbol{K}_{b,E}$. This choice is made because it is especially desirable to minimize the base pitch of Phoenix, as its manipulator arm is far better suited at tracking the pitch references due to its smaller inertia.

It is necessary to place a small threshold on $\boldsymbol{\sigma}$ in the adaptive gain update law (29) such that we have

$$\dot{\boldsymbol{k}}_1 = \begin{cases} \boldsymbol{\omega}_1 \sqrt{\dfrac{\boldsymbol{\gamma}_1}{2}}, & \text{if } |\boldsymbol{\sigma}| > \boldsymbol{\alpha}_m \\ \boldsymbol{0}, & \text{if } |\boldsymbol{\sigma}| \leq \boldsymbol{\alpha}_m \end{cases} \tag{34a}$$

$$\boldsymbol{k_2} = 2\boldsymbol{\epsilon}\boldsymbol{k_1} + \boldsymbol{\lambda} + 4\boldsymbol{\epsilon}^2, \tag{34b}$$

where the threshold $\boldsymbol{\alpha}_m > \boldsymbol{0}$ is a design parameter. This is done to prevent the gains from growing indefinitely, as $\boldsymbol{\sigma}$ will never be exactly zero due to numerical uncertainties. We choose this as $\boldsymbol{\alpha}_m = \left[5 \cdot 10^{-3}\boldsymbol{e}_6 \quad 5 \cdot 10^{-6}\boldsymbol{e}_n\right]^T$ for both vehicles, where $\boldsymbol{e}_i$ is a $1 \times i$ vector of ones. The gains of the dynamic controllers were chosen as shown in Table 1. The adaptive gain in (34a) was initialized as $\boldsymbol{k}_{1,P} = \left[500\boldsymbol{e}_6 \quad 10\boldsymbol{e}_{n_P}\right]^T$ and $\boldsymbol{k}_{1,E} = \boldsymbol{e}_{6+n_E}^T$ for Phoenix and Eelume, respectively.

Table 1. Dynamic controller gains

	Phoenix gains	**Eelume gains**
$\boldsymbol{\beta}$	$80\boldsymbol{e}_{6+n}^T$	$80\boldsymbol{e}_{6+n}^T$
$\boldsymbol{\lambda}$	$[0.1\boldsymbol{e}_6 \quad 10\boldsymbol{e}_n]^T$	$[0.01\boldsymbol{e}_3 \quad 5\boldsymbol{e}_3 \quad 10\boldsymbol{e}_{3+n}]^T$
$\boldsymbol{\epsilon}$	$\frac{\boldsymbol{\omega}_2}{2\boldsymbol{\omega}_1}\sqrt{\frac{\boldsymbol{\gamma}_2}{\boldsymbol{\gamma}_1}}\boldsymbol{e}_{6+n}^T$	$\frac{\boldsymbol{\omega}_2}{2\boldsymbol{\omega}_1}\sqrt{\frac{\boldsymbol{\gamma}_2}{\boldsymbol{\gamma}_1}}\boldsymbol{e}_{6+n}^T$
$\boldsymbol{\gamma}_1$	$\boldsymbol{e}_{6+n}^T$	$\boldsymbol{e}_{6+n}^T$
$\boldsymbol{\omega}_1$	$2\boldsymbol{e}_{6+n}^T$	$2\boldsymbol{e}_{6+n}^T$
$\boldsymbol{\gamma}_2$	$\begin{bmatrix} 4 \cdot 10^{-7}\boldsymbol{e}_3^T \\ 8 \cdot 10^{-9}\boldsymbol{e}_3^T \\ 4 \cdot 10^{-10}\boldsymbol{e}_n^T \end{bmatrix}$	$\begin{bmatrix} 4 \cdot 10^{-10}\boldsymbol{e}_3^T \\ 8 \cdot 10^{-9}\boldsymbol{e}_3^T \\ 4 \cdot 10^{-10}\boldsymbol{e}_n^T \end{bmatrix}$
$\boldsymbol{\omega}_2$	$2\boldsymbol{e}_{6+n}^T$	$2\boldsymbol{e}_{6+n}^T$
$\boldsymbol{\alpha}_m$	$[5 \cdot 10^{-3}\boldsymbol{e}_6 \quad 5 \cdot 10^{-6}\boldsymbol{e}_n]^T$	$[5 \cdot 10^{-3}\boldsymbol{e}_6 \quad 5 \cdot 10^{-6}\boldsymbol{e}_n]^T$

6.4 Simulation Results

In Figure 1 the reference trajectories for the operation point $\boldsymbol{P}_d$ and the virtual line $\boldsymbol{L}_d$ are shown along with the actual trajectories $\boldsymbol{P}$ and $\boldsymbol{L}$ resulting from the coordinated motion of the end-effector positions of Phoenix and Eelume. Figure 2 shows the orientation tracking of the end-effectors, given by the vector part of the quaternion, $\boldsymbol{\epsilon}_e$. Furthermore, Figure 3 shows the tracking of the second-highest prioritized task, i.e. the base attitude tracking. The performance of the lowest-priority task, joint limit avoidance, is shown in Figure 4. Figure 5 shows the sliding surfaces of Phoenix and Eelume. Finally, in Figure 6, the thruster forces and joint torques of Phoenix and Eelume are shown.

Figure 1 shows that there is an initial deviation in the virtual line. Specifically, we have chosen to initialize the robots such that the virtual line is 30cm away from the reference in x, y, and z in order to demonstrate the asymptotic convergence of the tracking error dynamics (33). After the initial transient, Figures 1 and 2 show that the end-effector configuration tasks are tracked well, with a maximum position error of 6mm. This error could be reduced even further by increasing the time constant in the reference models. However, the demands of any subsea task, be it inspection, intervention, or transportation, would be well satisfied in terms of precision. It can further be observed, through Figures 3 and 4, that the performance of the main task is achieved at a small expense of the less prioritized tasks. This behavior is intended, as the cooperation task holds the highest priority. Specifically, it can be seen that the base orientation and joint angles of both UVMSs deviate from their references due to the requirements of the end-effector configuration task at both reference steps. However, these deviations are minimized as a result of the effects of the base orientation task and the joint limit avoidance task, whenever the tasks are compatible. Moreover, we see that the base pitch angle of Phoenix converges to zero while its roll angle does not. This is due to Phoenix' kinematics; it is not redundant with respect to its end-effector roll angle. Hence, the end-effector configuration task and the base orientation task are not fully compatible in the case of Phoenix, and thus,

only the desired roll angle in the end-effector task will be tracked, due to its higher priority. The joint angles of both vehicles stay well within their mechanical limits. Both sliding surfaces in Figure 5 are consistently small, such that they stay within the threshold $\boldsymbol{\alpha}_m$ of (34a). The simulations thus confirm that the tracking errors converge asymptotically to zero when tasks are compatible, which supports the theoretical results. The thruster forces and joint torques in Figure 6 are physically feasible both in terms of amplitude and gradient. Indeed, Eelume uses at most 1/4 of its available thruster force and half its available joint torque. By size comparison between Phoenix and Eelume, it is clear that Phoenix' thruster forces and joint torques are also well within physical limits. There is little to no chattering in the control input, which can be attributed to the second-order degree of the SMC.

7. CONCLUSIONS

In this paper the cooperation between UVMSs in a heterogeneous team is considered. The proposed approach consists of a cooperative control scheme that generates state-dependent reference signals in order to achieve synchronization of the end-effectors, as well as kinematic and dynamic controllers that jointly make the UVMSs track these references. A task-priority controller is employed to leverage the kinematic redundancy of the robots. The dynamics of the systems are controlled by the AGSTA, which has strong robustness properties. A simulation study of a work-class UVMS and an AIAUV is performed, which shows the effectiveness of the proposed approach. Future work includes extending the cooperation scheme as well as conducting physical experiments.

REFERENCES

Antonelli, G. (2018). *Underwater Robots*. 4th ed. Springer.

Antonelli, G. (2009). Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. *IEEE Trans. Robotics*, 25(5), 985–994.

Borlaug, I.L.G., Pettersen, K.Y., and Gravdahl, J.T. (2020). The generalized super-twisting algorithm with adaptive gains. In *Proc. 2020 European Control Conference,* St. Petersburg, Russia, 1624–1631.

Borlaug, I.L., Pettersen, K., and Gravdahl, J. (2021). Comparison of two second-order sliding mode control algorithms for an articulated intervention AUV: Theory and experimental results. *Ocean Engineering*, 222, 108480.

Casalino, G., Caccia, M., Caselli, S., Melchiorri, C., Antonelli, G., Caiti, A., Indiveri, G., Cannata, G., Simetti, E., Torelli, S., Sperindè, A., Wanderlingh, F., Muscolo, G., Bibuli, M., Bruzzone, G., Zereik, E., Odetti, A., Spirandelli, E., Ranieri, A., and Cataldi, E. (2016). Underwater intervention robotics: An outline of the Italian national project MARIS. *Marine Technology Society Journal*, 50, 98–107.

Castillo, I., Fridman, L., and Moreno, J. (2017). Super-twisting algorithm in presence of time and state dependent perturbations. *Int. Journal of Control*, 1–24.

Chiacchio, P., Chiaverini, S., Sciavicco, L., and Siciliano, B. (1991). Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *The International Journal of Robotics Research*, 10(4), 410–425.

Chiaverini, S. (1997). Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3), 398–410.

Fossen, T.I. (1994). *Guidance and Control of Ocean Vehicles*. John Wiley & Sons, Ltd.

From, P., Gravdahl, J.T., and Pettersen, K.Y. (2014). *Vehicle-Manipulator Systems*. Springer-Verlag London.

Healey, A. and Lienard, D. (1993). Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE Journal of Oceanic Engineering*, 18(3), 327–339.

Levant, A. and Levantovsky, L. (1993). Sliding order and sliding accuracy in sliding mode control. *International Journal of Control*, 58, 1247–1263.

Liljebäck, P. and Mills, R. (2017). Eelume: A flexible and subsea resident IMR vehicle. In *Proc. OCEANS, Aberdeen, Scotland*, 1–4.

Mora-Aguilar, J., Carvajal, C.P., Sánchez, J.S., and Andaluz, V.H. (2018). Cooperative control of sliding mode for mobile manipulators. In *Proc. 10th Int. Conf. on Social Robotics,* Qingdao, China, 253–264.

Padir, T. (2005). Kinematic redundancy resolution for two cooperating underwater vehicles with on-board manipulators. In *2005 IEEE International Conference on Systems, Man and Cybernetics,* Waikoloa, Hawaii, USA.

Padir, T. and Koivo, A.J. (2003). Modeling of two underwater vehicles with manipulators on-board. In *Proc. IEEE International Conference on Systems, Man and Cybernetics,* Washington, DC, USA, 1359–1364.

Sarkar, N., Yuh, J., and Podder, T.K. (1999). Adaptive control of underwater vehicle-manipulator systems subject to joint limits. In *Proc. 1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* Kyongju, Korea, 142–147.

Schmidt-Didlaukies, H.M., Sørensen, A.J., and Pettersen, K.Y. (2018). Modeling of articulated underwater robots for simulation and control. In *2018 IEEE/OES Autonomous Underwater Vehicle Workshop*, 1–7.

Shtessel, Y.B., Edwards, C., Fridman, L., and Levant, A. (2014). *Sliding mode control and observation*. Birkhäuser, New York.

Shtessel, Y.B., Moreno, J.A., Plestan, F., Fridman, L.M., and Poznyak, A.S. (2010). Super-twisting adaptive sliding mode control: A Lyapunov design. In *Proc. 49th IEEE Conference on Decision and Control*, 5109–5113.

Simetti, E., Casalino, G., Manerikar, N., Sperindé, A., Torelli, S., and Wanderlingh, F. (2015). Cooperation between autonomous underwater vehicle manipulations systems with minimal information exchange. *OCEANS 2015,* Genova, Italy.

Simetti, E., Turetta, A., and Casalino, G. (2009). Distributed control and coordination of cooperative mobile manipulator systems. In *Distributed Autonomous Robotic Systems*. Springer.

Simetti, E. and Casalino, G. (2017). Manipulation and transportation with cooperative underwater vehicle manipulator systems. *IEEE Journal of Oceanic Engineering*, 42(4), 782–799.

Yoerger, D. and Slotine, J. (1985). Robust trajectory control of underwater vehicles. *IEEE Journal of Oceanic Engineering*, 10(4), 462–470.