Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 6696–6702

Robust Hierarchical Tracking Control of Vehicle-Manipulator Systems [★]

Jan Inge Dyrhaug [*] [1] Erling Tveter [*] [1]
Henrik M. Schmidt-Didlaukies [*] Erlend A. Basso [*]
Kristin Y. Pettersen [*] Jan Tommy Gravdahl [*]

[*] *Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway (e-mail: {jan.i.dyrhaug, erling.tveter, henrik.schmidt, erlend.a.basso, kristin.y.pettersen, jan.tommy.gravdahl}@ntnu.no).*

Abstract: For vehicle-manipulator systems (VMSs) to perform precise operations, a robust tracking control framework is required. Moreover, to exploit the platform's redundancy, it is desirable that the framework allows several tasks to be completed simultaneously. This work presents a robust hierarchical tracking controller for floating-base robots. In addition to providing uniform global asymptotic stability (UGAS) in free motion and allowing a total task dimension greater than the number of degrees of freedom (DoFs), the proposed controller includes a sliding mode effect and is proved to achieve UGAS even in the presence of bounded disturbances. Moreover, it is proved that uniform global ultimate boundedness (UGUB) is obtained if a continuous approximation of the sliding mode term is used. The stability results are proved mathematically, and are validated through simulations of an articulated intervention autonomous underwater vehicle (AIAUV).

Keywords: autonomous underwater vehicles, adaptive and robust control in marine systems, vehicle-manipulator systems, autonomous mobile robots

1. INTRODUCTION

The motivation for this paper was found in the field of underwater vehicle-manipulator systems (UVMSs). Today, underwater inspection, maintenance, and repair (IMR) operations are usually performed by human divers or remotely operated vehicles (ROVs). ROVs are preferred when the conditions pose a risk to human safety or the operation requires significant time spent underwater. They are, however, expensive to use. The main cost driver of IMR operations within the offshore energy industry is the large surface vessels from which ROVs are deployed (Liljebäck and Mills, 2017). Increasing the autonomy eliminates the need for these, and thus autonomous underwater vehicles (AUVs) can help reduce both the cost and the human risk of IMR operations. In particular, the snake-like articulated intervention-AUV (AIAUV) combines the hovering and intervention capabilities of ROVs with the favorable hydrodynamic properties of survey-AUVs, making it very versatile and a good candidate for autonomous IMR.

Robustness is an important feature of VMS control systems in general and is crucial when operating underwater, as the underwater environment entails significant disturbances and modeling uncertainties. Furthermore, as VMSs are redundant with respect to the payload pose, due to their mobile base and manipulator arm, a tracking control framework must be able to handle this redundancy. One way of doing so is by using a task-priority framework where the tasks define the behavior of all degrees of freedom (DoFs) at all times. This type of framework also has the advantage of enabling the VMS to complete several tasks at once, for instance by using the redundant DoFs to avoid collisions.

Faced with multiple, possibly conflicting tasks, a redundant robotic system needs a way of prioritizing when resolving redundancy. Task-priority methods are usually either based on null-space projections (Siciliano and Slotine, 1991) or optimization (Kanoun et al., 2011). The projection methods obtain strict priority, whereas optimization methods usually use weighting and thus lack the strict priority between tasks. However, there exist methods that combine strict priority between hierarchy levels and soft priority between tasks on the same level, e.g. Basso and Pettersen (2020). Optimization-based methods also allow for the incorporation of inequality tasks, which is a natural way of describing many important tasks like obstacle avoidance, joint limits, and contact constraints.

Task-priority control is faced with several challenges concerning singularities, both kinematic and algorithmic (Chiaverini, 1997), conflicting tasks, and stability analysis. Most methods lack stability proofs for the whole system and only show asymptotic stability of the tasks. Moreover, assuming non-singular Jacobian matrices in the proofs is common. Singularity-robustness is achieved by methods such as

[★] 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.

[1] Equal contribution.

10.1016/j.ifacol.2023.10.373

damped least-squares instead of the regular pseudo-inverse (Nakamura and Hanafusa, 1986), or numerical filtering (Maciejewski and Klein, 1988). Chiaverini (1997) also introduced a technique for two tasks that overcomes algorithmic singularities, which was extended to an arbitrary number of tasks by Antonelli (2009) which employs closed-loop inverse kinematics (CLIK) to prevent drift of the desired joint angles. Chiacchio et al. (1991) uses the transposed Jacobian instead of its pseudo-inverse.

While the hierarchical operational space formulation with full feedback linearization on each priority level gives exponential stability of the trajectory tracking error in theory, it is sensitive to modeling uncertainties, due to the cancellation of the dynamics, which also requires problematic feedback of external forces/torques. Instead, one can leave parts of the dynamics, e.g. the natural inertia, and utilize passivity in the controller design. An approach to tackle this in the regulation case is given by Ott et al. (2015), which proves asymptotic stability and nice passivity properties. This method was extended to tracking control in Dietrich and Ott (2020). However, both methods assume non-singular task Jacobians, and the tracking controller requires a total task dimension equal to the number of degrees of freedom. These requirements are removed in Garofalo and Ott (2020), while simultaneously proving the stronger property of UGAS. The controller in Garofalo and Ott (2020) has also been made adaptive in Garofalo et al. (2021). However, while guaranteeing asymptotic tracking of the tasks, the framework again requires that the total task dimension is equal to the number of DoFs. Moreover, strict priority between the tasks cannot generally be guaranteed.

Sæbø et al. (2022) provides an extension of the method of Dietrich and Ott (2020) to UVMSs. The method of Dietrich and Ott (2020) has recently also been improved by Wu et al. (2022a), proving uniform exponential stability, passivity during physical interaction and input-to-state stability, using external force feedback. Furthermore, Wu et al. (2022b) compared the method experimentally with both the hierarchical PD+ controller of Dietrich and Ott (2020) and feedback linearization.

In this work, we build on the control approach of Garofalo and Ott (2020) to obtain the strong properties of this method. We will adapt the method to VMSs, and also include hydrodynamic effects so that the class of systems includes UVMSs. The extended control law will include compensation for certain parts of the hydrodynamic effects; moreover, we introduce a sliding mode effect to achieve robustness to external disturbances and modeling uncertainties. We prove that the proposed control law provides uniform global asymptotic stability (UGAS) of the task error dynamics, even in the presence of external disturbances. Furthermore, a continuous approximation of the ideal sliding mode controller is proved to yield uniform global ultimate boundedness (UGUB). The theoretical results are validated in simulations, where the performances of the proposed control laws are also compared to control laws derived more directly from Garofalo and Ott (2020).

The paper is organized as follows: Section 2 presents a mathematical model of the class of VMSs, which also includes UVMSs, and the task specifications. The control design is given in Section 3, and the corresponding stability results are given in Section 4 and proved in Section 5. The results are validated through simulations in Section 6, where the different controllers are compared, before the conclusions and future work are presented in Section 7.

2. VMS MODEL AND TASK SPECIFICATIONS

The state of a VMS can be described by the pose of its base and the manipulator configuration. The pose and velocities of the robot's base are given by

$$\boldsymbol{\eta}^\top = \begin{bmatrix}\mathbf{p}^\top & \mathbf{q}^\top\end{bmatrix} \in \mathbb{R}^7 \text{ and } \boldsymbol{\nu}^\top = \begin{bmatrix}\mathbf{v}^\top & \boldsymbol{\omega}^\top\end{bmatrix} \in \mathbb{R}^6, \quad (1)$$

respectively, with $\mathbf{p} \in \mathbb{R}^3$ being the position in the inertial frame, $\mathbf{q} \in \mathbb{S}^3 := \{\mathbf{q} \in \mathbb{R}^4 : \mathbf{q}^\top \mathbf{q} = 1\}$ a unit quaternion describing the orientation relative to the inertial frame and $\mathbf{v}, \boldsymbol{\omega} \in \mathbb{R}^3$ the body-fixed linear and angular velocities, respectively. A VMS with n joints will have $d = 6 + n$ DoFs and its full state will be given by combining the state of the base (1) with the joint state $\boldsymbol{\theta}, \dot{\boldsymbol{\theta}} \in \mathbb{R}^n$:

$$\boldsymbol{\xi}^\top = \begin{bmatrix}\boldsymbol{\eta}^\top & \boldsymbol{\theta}^\top\end{bmatrix} \in \mathbb{R}^{d+1} \text{ and } \boldsymbol{\zeta}^\top = \begin{bmatrix}\boldsymbol{\nu}^\top & \dot{\boldsymbol{\theta}}^\top\end{bmatrix} \in \mathbb{R}^d. \quad (2)$$

Notice that the joint state $\boldsymbol{\theta}$ is not constrained by the base pose $\boldsymbol{\eta}$ given an obstacle-free environment.

The dynamic model of a VMS is (From et al., 2014, Ch. 8.2)

$$\dot{\boldsymbol{\xi}} = \mathbf{J}_a(\boldsymbol{\xi})\boldsymbol{\zeta} \quad (3a)$$

$$\mathbf{M}(\boldsymbol{\theta})\dot{\boldsymbol{\zeta}} + \mathbf{C}(\boldsymbol{\theta}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \mathbf{D}(\boldsymbol{\theta}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \mathbf{g}(\boldsymbol{\xi}) = \boldsymbol{\tau} + \mathbf{f}, \quad (3b)$$

in a body-fixed frame. Here, $\mathbf{J}_a$ is the Jacobian transforming body-fixed velocities $\boldsymbol{\zeta}$ to velocities in the inertial frame $\dot{\boldsymbol{\xi}}$ and $\mathbf{M}$ is the system's inertia matrix, i.e. the sum of rigid body mass and, in the case of a marine system, added mass caused by hydrodynamic effects. The Coriolis and centripetal force matrix is given by $\mathbf{C}$, which also includes added mass effects in the case of UVMSs, while $\mathbf{D}$ denotes the hydrodynamic damping or viscous friction matrix depending on whether the system is marine or terrestrial. The vector $\mathbf{g}$ represents the buoyancy and gravitational forces, $\boldsymbol{\tau}$ the wrench caused by the actuators, while $\mathbf{f}$ is the sum of the external wrench (not captured by $\mathbf{D}\boldsymbol{\zeta}$ or $\mathbf{g}$) and any modeling errors affecting the system.

It can be shown that for UVMSs with a suitable choice of $\mathbf{C}$, the following properties hold (Antonelli, 2018, Ch. 2.15):

$$\mathbf{M} = \mathbf{M}^\top > 0 \quad (4a)$$

$$\dot{\mathbf{M}} = \mathbf{C} + \mathbf{C}^\top \quad (4b)$$

$$\mathbf{z}^\top \mathbf{D} \mathbf{z} \geq 0, \quad (4c)$$

The same properties are known to also hold for terrestrial VMSs (Chung et al., 2016; De Luca and Book, 2016).

2.1 Task specifications

The tasks are specified as

$$\mathbf{x}_i = \mathbf{h}_i(\boldsymbol{\xi}) \quad (5)$$

$$\dot{\mathbf{x}}_i = \frac{\partial \mathbf{h}_i}{\partial \boldsymbol{\xi}}\dot{\boldsymbol{\xi}} = \frac{\partial \mathbf{h}_i}{\partial \boldsymbol{\xi}}\mathbf{J}_a\boldsymbol{\zeta} = \mathbf{J}_i(\boldsymbol{\xi})\boldsymbol{\zeta} \quad (6)$$

with $i \in \{1, 2, \ldots, r\}$ and $\mathbf{h}_i : \mathbb{R}^{d+1} \to \mathbb{R}^{m_i}$.

The tasks are stacked in the order of their priority, such that $\mathbf{x}_i$ has a higher priority than $\mathbf{x}_j$ for all $i < j$. The total task dimension is given by $m = \sum_{i=1}^{r} m_i$, and the whole stack is denoted as $\mathbf{x} = \begin{bmatrix}\mathbf{x}_1^\top \; \mathbf{x}_2^\top \; \dots \; \mathbf{x}_r^\top\end{bmatrix}^\top$. For every task, a desired trajectory $\mathbf{x}_{i,d}(t) \in \mathbb{R}^{m_i}$ is given, together with its first and second order time-derivatives, $\dot{\mathbf{x}}_{i,d}(t)$ and $\ddot{\mathbf{x}}_{i,d}(t)$. The desired trajectories are required to be compatible so that all tasks can be accomplished simultaneously.

3. CONTROLLER DESIGN

In this section we will build on the control approach of Garofalo and Ott (2020). We will extend this to the class of VMSs presented in Section 2, and we will include a sliding mode effect to achieve robustness to external disturbances and model errors. The closed-loop stability results and analysis will be presented in Sections 4 and 5.

To present the basics of the method that we will build on, and then to extend this, we need to first briefly present some definitions and propositions from Garofalo and Ott (2020). As in their work, the controller design relies on null-space projections to obtain dynamically decoupled task velocities that are consistent with the priority stack. To achieve this, the weighted Moore–Penrose inverse (WMPI) is used extensively. We use their simplified notation $\mathbf{A}^+ = \mathbf{A}^\dagger_{\mathbf{I},\mathbf{M}}$, where the weights are the identity matrix $\mathbf{I}$ and the inertia matrix $\mathbf{M}$, respectively.

3.1 Dynamically consistent Jacobian matrices

Each task Jacobian is projected onto the null-space of all the higher priority ones by the following construction:

Definition 1. (Garofalo and Ott (2020), Def. 1) The dynamically consistent Jacobian matrix $\bar{\mathbf{J}}_i$ is computed using the orthogonal projection matrix $\mathbf{P}_{i-1}$ and is defined as

$$\bar{\mathbf{J}}_i = \mathbf{J}_i \mathbf{P}_{i-1} \tag{7}$$

$$\mathbf{P}_i = \mathbf{P}_{i-1}(\mathbf{I} - \bar{\mathbf{J}}_i^+ \bar{\mathbf{J}}_i) \tag{8}$$

for $i \in \{1, \dots, r\}$, with $\mathbf{P}_0 = \mathbf{I}$ (the $d \times d$ identity matrix).

3.2 Change of coordinates

Here, the model will be represented by replacing the body velocities $\boldsymbol{\zeta}$ with decoupled task velocities $\mathbf{v}$.

Proposition 2. (Garofalo and Ott (2020), Prop. 4) Given the dynamically decoupled task velocities $\mathbf{v}_i = \bar{\mathbf{J}}_i \boldsymbol{\zeta}$, we have

$$\boldsymbol{\zeta} = \bar{\mathbf{J}}^+ \mathbf{v} \tag{9a}$$

$$\dot{\mathbf{x}} = \mathbf{T}\mathbf{v} \tag{9b}$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \cdots & \mathbf{0} \\ \mathbf{J}_2 \bar{\mathbf{J}}_1^+ & \mathbf{I} & \mathbf{0} \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots\cdots & \cdots \\ \mathbf{J}_r \bar{\mathbf{J}}_1^+ & \mathbf{J}_r \bar{\mathbf{J}}_2^+ & \cdots\cdots & \mathbf{I} \end{bmatrix} \tag{10}$$

is a lower block-triangular matrix with identities on the diagonal (and thus invertible) such that $\mathbf{J} = \mathbf{T}\bar{\mathbf{J}}$.

Using (9a) and pre-multiplying the VMS dynamics (3b) by $\bar{\mathbf{J}}^{+^\top}$, one obtains the dynamics in the new coordinates

$$\boldsymbol{\Lambda}\dot{\mathbf{v}} + (\boldsymbol{\Gamma} + \boldsymbol{\Delta})\mathbf{v} = \bar{\mathbf{J}}^{+^\top}(\boldsymbol{\tau} + \mathbf{f} - \mathbf{g}), \tag{11}$$

where $\boldsymbol{\Lambda} = \bar{\mathbf{J}}^{+^\top}\mathbf{M}\bar{\mathbf{J}}^+$, $\boldsymbol{\Gamma} = \bar{\mathbf{J}}^{+^\top}(\mathbf{C}\bar{\mathbf{J}}^+ + \mathbf{M}\dot{\bar{\mathbf{J}}}^+)$ and $\boldsymbol{\Delta} = \bar{\mathbf{J}}^{+^\top}\mathbf{D}\bar{\mathbf{J}}^+$.

The transformed inertia matrix is block-diagonal, $\boldsymbol{\Lambda} = \operatorname{diag}(\bar{\mathbf{J}}_i^{+^\top}\mathbf{M}\bar{\mathbf{J}}_i^+)$, which follows along the same lines as for the robot manipulator system in Garofalo and Ott (2020). Therefore, in these coordinates, there is no coupling between the tasks in the inertia term. Furthermore, using (4b) one can show that $\dot{\boldsymbol{\Lambda}} = \boldsymbol{\Gamma}_d + \boldsymbol{\Gamma}_d^\top$, where $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_d + \boldsymbol{\Gamma}_s$, $\boldsymbol{\Gamma}_d = \operatorname{diag}(\boldsymbol{\Gamma}_{d,i})$ is block-diagonal with $\boldsymbol{\Gamma}_{d,i} = \bar{\mathbf{J}}_i^{+^\top}(\mathbf{C}\bar{\mathbf{J}}_i^+ + \mathbf{M}\dot{\bar{\mathbf{J}}}_i^+)$ and $\boldsymbol{\Gamma}_s = -\boldsymbol{\Gamma}_s^\top$.

Moreover, we have that $\mathbf{z}^\top\boldsymbol{\Delta}\mathbf{z} \geq 0 \;\forall \mathbf{z}$ due to (4c). The matrix $\boldsymbol{\Delta}$ is then split into its block-diagonal and off-diagonal part, $\boldsymbol{\Delta}_d = \operatorname{diag}(\bar{\mathbf{J}}_i^{+\top}\mathbf{D}\bar{\mathbf{J}}_i^+)$ and $\boldsymbol{\Delta}_{nd} = \boldsymbol{\Delta} - \boldsymbol{\Delta}_d$, where $\mathbf{z}^\top\boldsymbol{\Delta}_d\mathbf{z} \geq 0 \;\forall \mathbf{z}$ as well.

3.3 Control law and closed-loop system

We now exploit the properties of the symmetric projector $\bar{\mathbf{J}}\bar{\mathbf{J}}^+ = \left(\bar{\mathbf{J}}\bar{\mathbf{J}}^+\right)^\top$.

Proposition 3. The projection matrix $\bar{\mathbf{J}}\bar{\mathbf{J}}^+$ satisfies

$$\bar{\mathbf{J}}\bar{\mathbf{J}}^+\boldsymbol{\Lambda} = \boldsymbol{\Lambda}, \quad \bar{\mathbf{J}}\bar{\mathbf{J}}^+\boldsymbol{\Delta}_d = \boldsymbol{\Delta}_d, \quad \bar{\mathbf{J}}\bar{\mathbf{J}}^+\boldsymbol{\Delta}_{nd} = \boldsymbol{\Delta}_{nd},$$
$$\bar{\mathbf{J}}\bar{\mathbf{J}}^+\boldsymbol{\Gamma} = \boldsymbol{\Gamma}, \quad \bar{\mathbf{J}}\bar{\mathbf{J}}^+\boldsymbol{\Gamma}_d = \boldsymbol{\Gamma}_d, \quad \bar{\mathbf{J}}\bar{\mathbf{J}}^+\boldsymbol{\Gamma}_s = \boldsymbol{\Gamma}_s.$$

A proof of this can be found in Garofalo and Ott (2020, Prop. 5). Their argument for $\boldsymbol{\Gamma}$ also holds for $\boldsymbol{\Delta}$.

The projected gains $\mathbf{K}_p$, $\mathbf{K}_d$, and the variables $\mathbf{v}_r$ and $\mathbf{s}$ are defined similarly as in Garofalo and Ott (2020): The gains are defined as $\mathbf{K}_p = \bar{\mathbf{J}}\bar{\mathbf{J}}^+\mathbf{K}'_p\bar{\mathbf{J}}\bar{\mathbf{J}}^+$ and $\mathbf{K}_d = \bar{\mathbf{J}}\bar{\mathbf{J}}^+\mathbf{K}'_d\bar{\mathbf{J}}\bar{\mathbf{J}}^+$ with block-diagonal matrices $\mathbf{K}'_p, \mathbf{K}'_d > \mathbf{0}$. The variables are the reference velocity $\mathbf{v}_r = \mathbf{v}_d - \mathbf{K}_p\tilde{\mathbf{x}}$, where $\mathbf{v}_d = \bar{\mathbf{J}}\bar{\mathbf{J}}^+\mathbf{T}^{-1}\dot{\mathbf{x}}_d$ and $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d$, while the sliding variable $\mathbf{s} = \mathbf{v} - \mathbf{v}_r$.

With this notation, we can adapt the control law of Garofalo and Ott (2020) to the VMS system (3):

$$\boldsymbol{\tau} = \mathbf{g} + \bar{\mathbf{J}}^\top(\boldsymbol{\Lambda}\dot{\mathbf{v}}_r + \boldsymbol{\Gamma}_d\mathbf{v}_r + \boldsymbol{\Gamma}_s\mathbf{v} - \mathbf{K}_d\mathbf{s}). \tag{12}$$

To achieve better tracking and guaranteed stability, we will in addition compensate for selected parts of the hydrodynamic effects. We therefore propose the control law

$$\boldsymbol{\tau} = \mathbf{g} + \bar{\mathbf{J}}^\top[\boldsymbol{\Lambda}\dot{\mathbf{v}}_r + (\boldsymbol{\Gamma}_d + \boldsymbol{\Delta}_d)\mathbf{v}_r + (\boldsymbol{\Gamma}_s + \boldsymbol{\Delta}_{nd})\mathbf{v} - \mathbf{K}_d\mathbf{s}]. \tag{13}$$

These two controllers (12) and (13) will be referred to as the *standard* and *hydrodynamic* controller, respectively. As in Garofalo and Ott (2020), one can replace $\mathbf{v}$ with $\mathbf{v}_r$ in the control laws without changing the stability results; however, strict decoupling of tasks is not ensured in that case.

As discussed in the Introduction, robustness is an important feature of VMS control systems in general, and is crucial when operating underwater. Garofalo and Ott (2020) did not consider external forces and model uncertainty. In the following, we seek to enhance the robustness by introducing a sliding mode effect in the control law. Specifically, we introduce the sliding mode term $\bar{\mathbf{J}}^\top\mathbf{K}_s\operatorname{sgn}(\mathbf{s})$ into (13), and propose the following *sliding mode controller (SMC)*:

$$\begin{aligned}\boldsymbol{\tau} = \mathbf{g} + \bar{\mathbf{J}}^\top[\boldsymbol{\Lambda}\dot{\mathbf{v}}_r + (\boldsymbol{\Gamma}_d + \boldsymbol{\Delta}_d)\mathbf{v}_r + (\boldsymbol{\Gamma}_s + \boldsymbol{\Delta}_{nd})\mathbf{v} \\ - \mathbf{K}_d\mathbf{s} - \mathbf{K}_s\operatorname{sgn}(\mathbf{s})],\end{aligned} \tag{14}$$

where $\operatorname{sgn}(\cdot)$ is the element-wise sign function, $\mathbf{K}_s = \bar{\mathbf{J}}\bar{\mathbf{J}}^+\mathbf{K}'_s\bar{\mathbf{J}}\bar{\mathbf{J}}^+$ is sufficiently large, and $\mathbf{K}'_s > 0$ is chosen as diagonal. In Section 5 we will prove that the resulting control law achieves robustness with respect to bounded external forces and model uncertainty.

Table 1. The four different control laws.

Name	**Expression**	**Description**
Standard	$\boldsymbol{\tau} = \mathbf{g} + \bar{\mathbf{J}}^\top(\boldsymbol{\Lambda}\dot{\mathbf{v}}_r + \boldsymbol{\Gamma}_d\mathbf{v}_r + \boldsymbol{\Gamma}_s\mathbf{v} - \mathbf{K}_d\mathbf{s})$ (12)	Garofalo and Ott (2020), extended to VMS.
Hydrodynamic	$\boldsymbol{\tau} = \mathbf{g} + \bar{\mathbf{J}}^\top(\boldsymbol{\Lambda}\dot{\mathbf{v}}_r + (\boldsymbol{\Gamma}_d + \boldsymbol{\Delta}_d)\mathbf{v}_r + (\boldsymbol{\Gamma}_s + \boldsymbol{\Delta}_{nd})\mathbf{v} - \mathbf{K}_d\mathbf{s})$ (13)	Adds compensation for hydrodyn. damping.
SMC	$\boldsymbol{\tau} = \mathbf{g} + \bar{\mathbf{J}}^\top(\boldsymbol{\Lambda}\dot{\mathbf{v}}_r + (\boldsymbol{\Gamma}_d + \boldsymbol{\Delta}_d)\mathbf{v}_r + (\boldsymbol{\Gamma}_s + \boldsymbol{\Delta}_{nd})\mathbf{v} - \mathbf{K}_d\mathbf{s} - \mathbf{K}_s\,\mathrm{sgn}(\mathbf{s}))$ (14)	Adds a first-order sliding mode term.
CSMA	$\boldsymbol{\tau} = \mathbf{g} + \bar{\mathbf{J}}^\top(\boldsymbol{\Lambda}\dot{\mathbf{v}}_r + (\boldsymbol{\Gamma}_d + \boldsymbol{\Delta}_d)\mathbf{v}_r + (\boldsymbol{\Gamma}_s + \boldsymbol{\Delta}_{nd})\mathbf{v} - \mathbf{K}_d\mathbf{s} - \mathbf{K}_s\,\mathrm{sat}_\varepsilon(\mathbf{s}))$ (16)	Replaces sgn with sat.

A discontinuous control law may, however, give rise to severe chattering and may thus be undesirable in practice. Using a continuous approximation of the sign function, e.g. the saturation function

$$\mathrm{sat}_\varepsilon(x) := \begin{cases} \mathrm{sgn}(x) & \text{if } |x| > \varepsilon, \\ x/\varepsilon & \text{otherwise,} \end{cases} \tag{15}$$

instead of $\mathrm{sgn}(\cdot)$ on each element of $\mathbf{s}$ in the sliding mode term, the controller becomes continuous:

$$\begin{aligned} \boldsymbol{\tau} = \mathbf{g} + \bar{\mathbf{J}}^\top[\boldsymbol{\Lambda}\dot{\mathbf{v}}_r + (\boldsymbol{\Gamma}_d + \boldsymbol{\Delta}_d)\mathbf{v}_r + (\boldsymbol{\Gamma}_s + \boldsymbol{\Delta}_{nd})\mathbf{v} \\ - \mathbf{K}_d\mathbf{s} - \mathbf{K}_s\,\mathrm{sat}_\varepsilon(\mathbf{s})]. \end{aligned} \tag{16}$$

This mitigates the problem of chattering. In the following we call (16) the *continuous sliding mode approximation (CSMA)* controller. All the proposed control laws are summarized in Table 1.

Defining $\tilde{\mathbf{v}} = \mathbf{v} - \mathbf{v}_d$, one can write

$$\tilde{\mathbf{v}} = -\mathbf{K}_p\tilde{\mathbf{x}} + \mathbf{s}, \tag{17}$$

which is unaffected by the choice of controller.

Applying (13) to (11) yields the closed-loop $\mathbf{s}$-dynamics

$$\boldsymbol{\Lambda}\dot{\mathbf{s}} = -(\boldsymbol{\Gamma}_d + \boldsymbol{\Delta}_d + \mathbf{K}_d)\mathbf{s} + \bar{\mathbf{J}}^{+^\top}\mathbf{f}, \tag{18}$$

where the properties in Proposition 3 have been used. With (14), the closed-loop $\mathbf{s}$-dynamics become

$$\boldsymbol{\Lambda}\dot{\mathbf{s}} = -(\boldsymbol{\Gamma}_d + \boldsymbol{\Delta}_d + \mathbf{K}_d)\mathbf{s} + \bar{\mathbf{J}}^{+^\top}\mathbf{f} - \mathbf{K}_s\,\mathrm{sgn}(\mathbf{s}), \tag{19}$$

and with (16), sgn is replaced by sat:

$$\boldsymbol{\Lambda}\dot{\mathbf{s}} = -(\boldsymbol{\Gamma}_d + \boldsymbol{\Delta}_d + \mathbf{K}_d)\mathbf{s} + \bar{\mathbf{J}}^{+^\top}\mathbf{f} - \mathbf{K}_s\,\mathrm{sat}_\varepsilon(\mathbf{s}). \tag{20}$$

4. STABILITY RESULTS

In this section we will present the main results of this paper. The proofs are given in Section 5.

Assumption 4. The total task dimension is not smaller than the number of DoFs, i.e., $m \geq d = n + 6$.

Remark: This merely requires that the specified tasks completely define the behavior of the VMS.

Assumption 5. The inertia matrix is bounded by

$$0 < \alpha_{\mathbf{M}} \leq \|\mathbf{M}\| \leq \beta_{\mathbf{M}} < \infty,$$

where $\alpha_{\mathbf{M}}$ and $\beta_{\mathbf{M}}$ are positive constants.

Remark: This is always true for terrestrial VMSs with non-singular formulations and only revolute joints.

Assumption 6. $\bar{\mathbf{J}}$ is always full rank, i.e., $\mathrm{rank}(\bar{\mathbf{J}}) = d$.

Assumption 7. Each $\bar{\mathbf{J}}_i$ is bounded and its nonzero singular values are bounded away from zero.

Remark: The boundedness of each $\bar{\mathbf{J}}_i$ is not very restrictive, as they are all bounded when the manipulator consists solely of revolute joints and the position tasks are specified as $\mathbf{x}_i(\boldsymbol{\xi}) = \mathbf{p}$. Furthermore, with Assumptions 5 and 7, each $\bar{\mathbf{J}}_i^+$ is bounded as well. Thus, $\boldsymbol{\xi} \to \mathbf{T}(\boldsymbol{\xi})$ is also bounded. In numerical implementation, singular values below a certain threshold are set to zero.

Theorem 8. Given Assumption 4 with $m = d$, and Assumptions 5, 6, and 7. Then, the closed-loop system (17, 18) with $\mathbf{f} = \mathbf{0}$, obtained by applying (13) to (11), is UGAS. Similarly, (14) and (16) obtain UGAS systems when $\mathbf{f} = \mathbf{0}$, but also obtain UGAS and UGUB systems, respectively, in the more general case of a bounded $\mathbf{f}$.

Corollary 9. Given Assumptions 4, 5, 6, and 7, with the i-th task being either singularity-free or having $\bar{\mathbf{J}}_i = \mathbf{0}$. Then, the closed-loop system (17, 18) with $\mathbf{f} = \mathbf{0}$, obtained by applying (13) to (11), is UGAS. Correspondingly, (14) and (16) obtain UGAS systems with $\mathbf{f} = \mathbf{0}$, and UGAS and UGUB systems, respectively, in the case of a bounded $\mathbf{f}$.

Corollary 10. If only one-dimensional tasks are designed for the system (3b), with the last d tasks being independent configuration tasks, and Assumptions 5 and 7 are satisfied, the closed-loop system with (13), (14), or (16) is UGAS if $\mathbf{f} = \mathbf{0}$, and UGAS or UGUB, respectively, for (14) and (16) if $\mathbf{f}$ is bounded.

5. STABILITY ANALYSIS

Proof of Theorem 8: Similarly as in Garofalo and Ott (2020), the system (17, 18) can be considered as the feedforward interconnection of two subsystems. The second subsystem (17) is itself a feedforward interconnection of subsystems. This becomes clear when writing it as

$$\dot{\tilde{\mathbf{x}}} = -\mathbf{T}\mathbf{K}_p\tilde{\mathbf{x}} + \mathbf{T}\mathbf{s}, \tag{21}$$

due to the triangular structure of $\mathbf{T}$. Note that $\bar{\mathbf{J}}\bar{\mathbf{J}}^+ = \mathbf{I}$ is used in this transition, which is true under the assumptions of Theorem 8. We will now show that, for each control law (13), (14), and (16), the whole task error dynamics $(\tilde{\mathbf{x}}, \dot{\tilde{\mathbf{x}}}) = (\mathbf{0}, \mathbf{0})$ is UGAS when $\mathbf{s} = \mathbf{0}$ is a UGAS equilibrium point of (18), (19), and (20), respectively. Furthermore, we will show that $(\tilde{\mathbf{x}}, \dot{\tilde{\mathbf{x}}})$ is UGUB when $\mathbf{s}$ is UGUB.

The triangular structure of (21) means that the dynamics of each subsystem $\tilde{\mathbf{x}}_i$ only depend on $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_i$ and $\mathbf{s}_1, \ldots, \mathbf{s}_i$. For example, $\dot{\tilde{\mathbf{x}}}_1 = -\mathbf{K}_{p_1}\tilde{\mathbf{x}}_1 + \mathbf{s}_1$ and $\dot{\tilde{\mathbf{x}}}_2 = -\mathbf{K}_{p_2}\tilde{\mathbf{x}}_2 - \mathbf{J}_2\bar{\mathbf{J}}_1^+\mathbf{K}_{p_1}\tilde{\mathbf{x}}_1 + \mathbf{J}_2\bar{\mathbf{J}}_1^+\mathbf{s}_1 + \mathbf{s}_2$. Since each block $\mathbf{K}_{p_i}$ is positive definite and constant, it follows from Khalil (2002, Lemma 4.6) that the first system, $\tilde{\mathbf{x}}_1$, is ISS with $\mathbf{s}$ as input. Combined with the assumption that $\mathbf{s} = \mathbf{0}$ is UGAS, we can conclude that $(\tilde{\mathbf{x}}_1, \mathbf{s}) = (\mathbf{0}, \mathbf{0})$ is UGAS (Khalil, 2002, Lemma 4.7). Next, the $\tilde{\mathbf{x}}_2$-system is ISS with $(-\mathbf{J}_2\bar{\mathbf{J}}_1^+\mathbf{K}_{p_1}\tilde{\mathbf{x}}_1 + \mathbf{J}_2\bar{\mathbf{J}}_1^+\mathbf{s}_1 + \mathbf{s}_2)$ as input. With Assumptions 5 and 7 we can conclude that $(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \mathbf{s}) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ is UGAS. This argument can be repeated for all the tasks, resulting in the conclusion that $(\tilde{\mathbf{x}}, \dot{\tilde{\mathbf{x}}}) = (\mathbf{0}, \mathbf{0})$ is UGAS when $\mathbf{s} = \mathbf{0}$ is UGAS. (The last transition holds since $\mathbf{s} = \mathbf{T}^{-1}\dot{\tilde{\mathbf{x}}} + \mathbf{K}_p\tilde{\mathbf{x}}$ when $\bar{\mathbf{J}}\bar{\mathbf{J}}^+ = \mathbf{I}$). The same line of reasoning holds when $\mathbf{s}$ is UGUB; by the definition of ISS (Khalil, 2002, Def. 4.7), a cascaded system consisting of an ISS system perturbed by a UGUB system, is UGUB.

Now that we have established the UGAS of the equilibrium point $\tilde{\mathbf{x}} = \mathbf{0}$ of the second subsystem (21) when $\mathbf{s} = \mathbf{0}$, we consider the first subsystem with state $\mathbf{s}$. Firstly, we consider the closed-loop system with the *hydrodynamic* control law (13), described by (18). For the first subsystem, (18), consider the function

$$V_s = \frac{1}{2}\mathbf{s}^\top \mathbf{\Lambda}\mathbf{s}. \tag{22}$$

Under the assumptions of the theorem, there exist constants $c_1 > 0$ and $c_2 > 0$ such that

$$c_1\|\mathbf{s}\|^2 \leq V_s \leq c_2\|\mathbf{s}\|^2 \tag{23}$$

holds for all configurations $\boldsymbol{\xi} \in \mathbb{R}^{d+1}$ and thus for all t.

Differentiating V_s along the flow of (18) yields

$$\dot{V}_s = -\mathbf{s}^\top(\mathbf{K}_d + \mathbf{\Delta}_d)\mathbf{s} + \mathbf{s}^\top\bar{\mathbf{J}}^{+\top}\mathbf{f}, \tag{24}$$

where $-\mathbf{s}^\top(\mathbf{K}_d+\mathbf{\Delta}_d)\mathbf{s} < 0 \quad \forall \mathbf{s} \neq \mathbf{0}$. Thus, in the absence of external forces and model uncertainty, the first subsystem, (18) is UGAS.

If instead $\mathbf{f} \neq \mathbf{0}$, $\dot{V}_s < 0$ can no longer be guaranteed using (13). Next, we consider the stability of the closed-loop system (19) where the *SMC* control law (14) is applied. We then get

$$\dot{V}_s = -\mathbf{s}^\top(\mathbf{K}_d + \mathbf{\Delta}_d)\mathbf{s} + \mathbf{s}^\top\left(\bar{\mathbf{J}}^{+\top}\mathbf{f} - \mathbf{K}_s\,\mathrm{sgn}(\mathbf{s})\right), \tag{25}$$

and sgn(·) is replaced by sat(·) in the case of *CSMA* (16). The last term of $\dot{V}_s$ can then be bounded as follows:

$$\begin{aligned}\mathbf{s}^\top\left(\bar{\mathbf{J}}^{+\top}\mathbf{f} - \mathbf{K}_s\,\mathrm{sgn}(\mathbf{s})\right) &= \sum_{i=1}^{r}\mathbf{s}_i^\top\left(\bar{\mathbf{J}}_i^{+\top}\mathbf{f} - \mathbf{K}_{s_i}\,\mathrm{sgn}(\mathbf{s}_i)\right)\\ &\leq \sum_{i=1}^{r}|\mathbf{s}_i|^\top\left(|\bar{\mathbf{J}}_i^{+\top}\mathbf{f}| - \mathrm{diag}(\mathbf{K}_{s_i})\right),\end{aligned}$$

where $|\cdot|$ is the element-wise absolute value, $\mathbf{K}_{s_i}$ denotes each block along the diagonal of $\mathbf{K}_s$, which has a block-diagonal structure, and $\mathbf{s}_i$ are the corresponding elements of $\mathbf{s}$. Since $\bar{\mathbf{J}}\bar{\mathbf{J}}^+ = \mathbf{I}$, we have $\mathbf{K}_s = \mathbf{K}'_s$.

In conclusion, we obtain

$$\dot{V}_s \leq -\mathbf{s}^\top(\mathbf{K}_d + \mathbf{\Delta}_d)\mathbf{s} - \kappa\|\mathbf{s}\|_1 \tag{26}$$

when $\mathrm{diag}(\mathbf{K}'_{s_i}) - |\bar{\mathbf{J}}_i^{+\top}\mathbf{f}| \geq \kappa\mathbf{1}$ for some constant $\kappa > 0$. The equilibrium $\mathbf{s} = \mathbf{0}$ is UGAS, with the first term in (26) giving exponential stability and the second giving finite time convergence towards the sliding surface, by the comparison lemma (Khalil, 2002, Lemma 3.4). Due to the previously shown structure of (21), by Khalil (2002, Lemma 4.7) and the definition of $\mathbf{s}$, we can conclude that $(\tilde{\mathbf{x}}, \dot{\tilde{\mathbf{x}}}) = (\mathbf{0}, \mathbf{0})$ is UGAS.

Using *CSMA* (16), which gives (20), the conditions of Khalil (2002, Theorem 4.18) hold with $D = \mathbb{R}^m$, $\alpha_1(r) = c_1r^2$, $\alpha_2(r) = c_2r^2 \in \mathcal{K}_\infty$ and any initial state $\mathbf{s}(t_0)$. Hence, $\mathbf{s}$ is UGUB with $(c_2/c_1)\varepsilon$ as the ultimate bound, and $(\tilde{\mathbf{x}}, \dot{\tilde{\mathbf{x}}})$ is UGUB due to the structure of (21). It can be shown that the ultimate bound on $\tilde{\mathbf{x}}$ decreases with decreasing ε and $\|\mathbf{T}\|$ and increasing $\lambda_{\min}(\mathbf{K}_p)$. The bound on $\dot{\tilde{\mathbf{x}}}$ has the same dependencies, in addition to decreasing with decreasing $\lambda_{\max}(\mathbf{K}_p)$. □

Proof of Corollary 9: As in Garofalo and Ott (2020), we utilize that $\bar{\mathbf{J}}_i$ is assumed to be either full row rank or equal to the zero matrix. Hence, the tasks with $\bar{\mathbf{J}}_i = \mathbf{0}$ yield identities of the form $0 = 0$ in the dynamics, and can be omitted from further analysis. The proof then reduces to that of Theorem 8. □

Proof of Corollary 10: The proof is identical to that of Garofalo and Ott (2020, Cor. 1): The requirement on the last d tasks implies that Assumptions 4 and 6 are satisfied. With scalar tasks, the only way $\bar{\mathbf{J}}_i$ can be singular is if $\bar{\mathbf{J}}_i = 0$. The result therefore follows from Corollary 9. □

6. SIMULATIONS

The proposed controllers were simulated on an AIAUV with four z- and four y-revolute joints, meaning the system had 14 DoFs in total. Furthermore, the AIAUV was equipped with seven thrusters producing up to 40 N of force each, and a motor in every joint producing up to 10 Nm of torque each. The simulations were run using an ode4 solver in MATLAB/Simulink with 1 ms time steps, and when performing the singular value decomposition needed to compute the WMPIs, the singular values were set to zero if they were below 0.01. To obtain a fair comparison, the controller gains were tuned so that the control inputs were of similar size. The controller gains were set to $\mathbf{K}'_p = 1.2\mathbf{I}$ and $\mathbf{K}'_d = 200\mathbf{I}$ for (12) and (13), and $\mathbf{K}'_p = \mathbf{K}'_d = 3\mathbf{I}$ and $\mathbf{K}'_s = 5\mathbf{I}$ for (14) and (16), while the ε in (15) was set to 0.01. During the simulations, only the linear part of the damping matrix $\mathbf{D}$ was assumed known to the controller.

In decreasing priority, the tasks provided to the controllers were the end effector pose, the base pose, and each individual joint angle. The robot was initially in a "C-shape", with each z-revolute joint having a 30-degree angle, and all y-revolute joints having zero-degree angles. The tasks were for the end effector to smoothly move forward and backward 20 cm along the inertial x-axis, while the rest of the end effector pose, along with the base pose and the joint angles, were to be kept constant.

To test the robustness to modeling errors, the control laws were implemented with a 40% error in the inertia and Coriolis matrices ($\mathbf{M}$ and $\mathbf{C}$ from (3b)). Additionally, a constant disturbance $\mathbf{f} = \mathbf{1}$, also from (3b), was added to the simulation to test the robustness to other disturbances. Note that even if $\mathbf{f}$ is constant, the effect on the closed-loop system is not, due to the change in $\bar{\mathbf{J}}^{+\top}$, see e.g. (18). The initial positions of the end effector and base were chosen to have a 10 cm offset with respect to the reference along each of the inertial axes.

In Fig. 1, the pose and joint angle task errors are shown. Only the first two joint angle tasks are performed, as the pose tasks use up all but two of the system's DoFs. The large errors of the rest of the joint angle tasks are therefore expected. After about four seconds, the *SMC* (14) and *CSMA* (16) control laws yield better tracking of the desired position trajectory than the *standard* (12) and *hydrodynamic* (13) control laws do, while the orientation and active joint angle task errors are kept lower even during the transient period. In accordance with Corollary 9, the errors converge (close) to zero when using the *SMC* (14) controller, are ultimately bounded when using the *CSMA* (16) controller, and have more significant stationary errors using the *standard* (12) or *hydrodynamic* (13) control laws. The mean, the maximum and the standard deviation of the

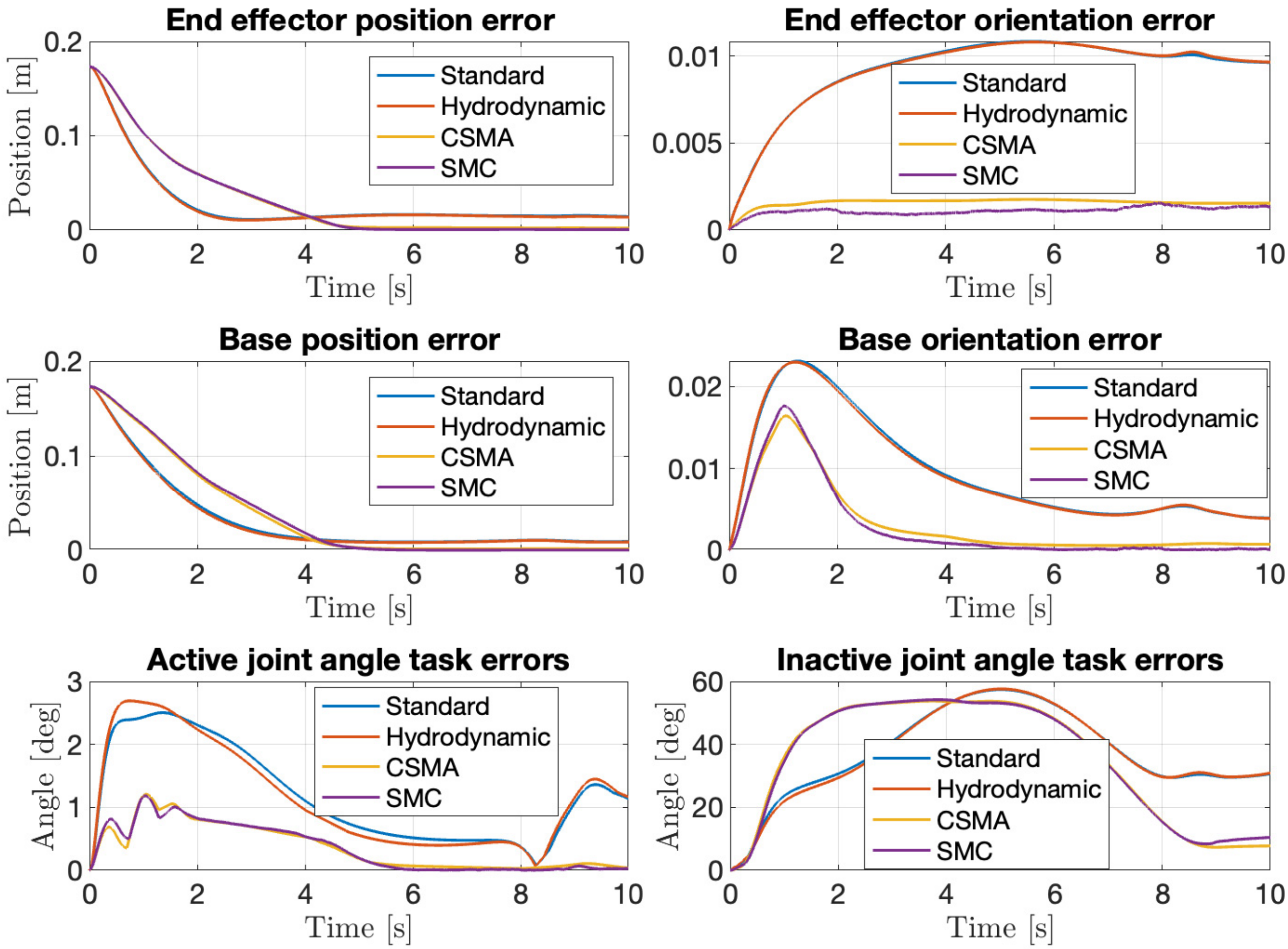

Fig. 1. Two-norm of the pose task errors, and one-norm of the active and inactive joint angle task errors for the different control laws. The error quaternion's vector part has been used for the orientation error.

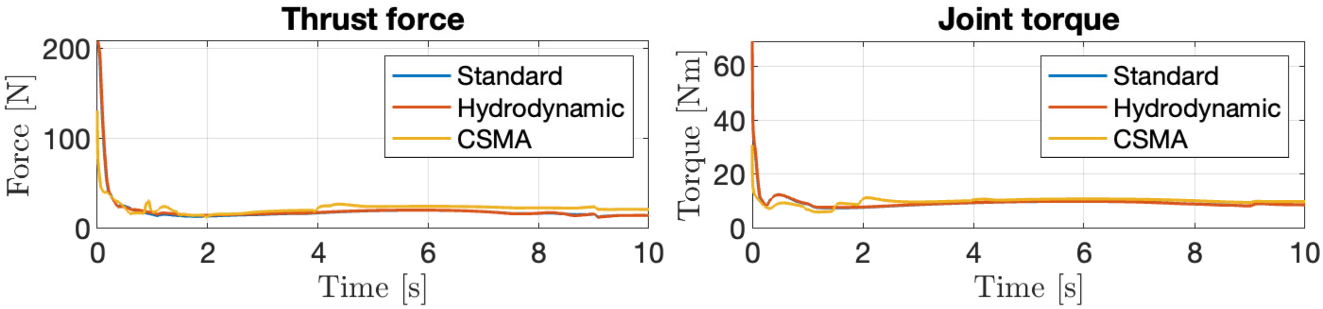

Fig. 2. One-norm of the applied thruster forces and joint torques for three of the control laws.

steady-state end effector pose tracking errors are given in Table 6.

The applied thruster forces and joint torques when using the *standard* (12), *hydrodynamic* (13), and *CSMA* (16) control laws are shown in Fig. 2. The control inputs when using the *SMC* (14) controller suffer heavily from chattering and are therefore not visualized. Furthermore, with $\sum_k u[k]^2 \Delta t$ used as a proxy for the energy consumption, where u are the control inputs and Δt is the simulation time step, the thrusters and joint motors are found to consume energy an order of magnitude larger when using the *SMC* (14) controller instead of one of the other proposed control laws over the time of the simulation.

As seen in Fig. 2, the control inputs when using the *standard* (12), *hydrodynamic* (13) and *CSMA* (16) control laws are relatively similar, apart from the initial ones, where those yielded by the *standard* (12) and *hydrodynamic* (13) control laws are significantly larger than those yielded by the *CSMA* (16) controller. The difference in energy consumption is less than 5%, apart from the energy consumption of the joint motors when using the *CSMA* (16) controller, which is about 13% higher than when using the *standard* (12) control law.

Table 2. The mean, maximal (Max), and standard deviation (Std) of the steady-state end effector pose error vectors' two norms using the different control laws. The error quaternion's vector part has been used for orientation, and steady-state is said to be achieved after a transient period of five seconds.

Controller		**Position**			**Orientation**		
		Mean	Max	Std	Mean	Max	Std
Standard	(12)	1.56 cm	1.65 cm	0.60 mm	1.03×10^{-2}	1.09×10^{-2}	4.16×10^{-4}
Hydrodynamic	(13)	1.50 cm	1.61 cm	0.78 mm	1.03×10^{-2}	1.08×10^{-2}	3.81×10^{-4}
SMC	(14)	**0.07 cm**	**0.16 cm**	0.15 mm	$\mathbf{0.13 \times 10^{-2}}$	$\mathbf{0.16 \times 10^{-2}}$	1.25×10^{-4}
CSMA	(16)	0.24 cm	0.30 cm	0.23 mm	0.16×10^{-2}	0.18×10^{-2}	0.86×10^{-4}

7. CONCLUSIONS AND FUTURE WORK

This paper has extended the control approach in Garofalo and Ott (2020) to VMSs, including UVMSs. In addition to including hydrodynamic compensation in the controller, we have added a sliding mode effect to improve the robustness. The closed-loop error dynamics are shown to be UGAS even in the presence of bounded disturbances, and UGUB with a continuous approximation of the sliding mode term. The methods are validated and compared through simulations of an AIAUV. The simulations show that the extended control law with the sliding mode effect increases the robustness to disturbances and model errors, yielding better tracking performance than without the sliding mode term. We also find that the continuous approximation of the sliding mode term seems to be a better choice in practice as it yields more achievable control inputs. In future work, the method should be tested in physical experiments.

ACKNOWLEDGEMENTS

We would like to thank Ph.D. candidate Marianna Wrzos-Kaminska for helpful discussions and advice.

REFERENCES

Antonelli, G. (2009). Stability Analysis for Prioritized Closed-Loop Inverse Kinematic Algorithms for Redundant Robotic Systems. *IEEE Trans. Robot.*, 25(5).

Antonelli, G. (2018). *Underwater Robots*, volume 123 of *Springer Tracts in Advanced Robotics*. Springer International Publishing, Cham.

Basso, E.A. and Pettersen, K.Y. (2020). Task-Priority Control of Redundant Robotic Systems using Control Lyapunov and Control Barrier Function based Quadratic Programs. In *Proc. IFAC World Cong.* Berlin, Germany.

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. *Int. J. Robotics Research*, 10(4), 410–425.

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

Chung, W.K., Fu, L.C., and Kröger, T. (2016). Motion Control. In B. Siciliano and O. Khatib (eds.), *Springer Handbook of Robotics*. Springer International Publishing.

De Luca, A. and Book, W.J. (2016). Robots with Flexible Elements. In B. Siciliano and O. Khatib (eds.), *Springer Handbook of Robotics*. Springer International Publishing.

Dietrich, A. and Ott, C. (2020). Hierarchical Impedance-Based Tracking Control of Kinematically Redundant Robots. *IEEE Trans. on Robotics*, 36(1), 204–221.

From, P.J., Gravdahl, J.T., and Pettersen, K.Y. (2014). *Vehicle-Manipulator Systems: Modeling for Simulation, Analysis, and Control.* Advances in Industrial Control. Springer London, London.

Garofalo, G. and Ott, C. (2020). Hierarchical Tracking Control With Arbitrary Task Dimensions: Application to Trajectory Tracking on Submanifolds. *IEEE Robotics and Automation Letters*, 5(4), 6153–6160.

Garofalo, G., Wu, X., and Ott, C. (2021). Adaptive Passivity-Based Multi-Task Tracking Control for Robotic Manipulators. *IEEE Robotics and Automation Letters*, 6(4), 7129–7136.

Kanoun, O., Lamiraux, F., and Wieber, P.B. (2011). Kinematic Control of Redundant Manipulators: Generalizing the Task-Priority Framework to Inequality Task. *IEEE Trans. on Robotics*, 27(4), 785–792.

Khalil, H. (2002). *Nonlinear Systems.* Pearson Education. Prentice Hall, 3rd edition.

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

Maciejewski, A.A. and Klein, C.A. (1988). Numerical filtering for the operation of robotic manipulators through kinematically singular configurations. *J. Robotic Systems*, 5(6), 527–552.

Nakamura, Y. and Hanafusa, H. (1986). Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control. *J. Dynamic Syst., Meas., and Control*, 108(3), 163–171.

Ott, C., Dietrich, A., and Albu-Schäeffer, A. (2015). Prioritized Multi-Task Compliance Control of Redundant Manipulators. *Automatica*, 53, 416–423.

Sæbø, B.K., Pettersen, K.Y., and Gravdahl, J.T. (2022). Robust Task-Priority Impedance Control for Vehicle-Manipulator Systems. In *Proc. IEEE Conf. Control Technol. and Appl.* Trieste, Italy.

Siciliano, B. and Slotine, J.J. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *Proc. 5th Int. Conf. Adv. Robot.* Pisa, Italy.

Wu, X., Ott, C., Albu-Schäffer, A., and Dietrich, A. (2022a). Passive Decoupled Multitask Controller for Redundant Robots. *IEEE Trans. on Control Syst. Technol.*, 1–16.

Wu, X., Ott, C., and Dietrich, A. (2022b). A Comparative Experimental Study of Multi-Tasking Tracking and Interaction Control on a Torque-Controlled Humanoid Robot. In *Proc. American Control Conf.* Atlanta, USA.