Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 58-20 (2024) 347–353

Power-Based Safety Constraint for Redundant Robotic Manipulators ⋆

Erling Tveter * **Kristin Y. Pettersen** * **Jan Tommy Gravdahl** *

* *Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway (e-mail: {erling.tveter, kristin.y.pettersen, jan.tommy.gravdahl}@ntnu.no).*

Abstract: A robot making or losing contact with its environment will experience a sudden change in its dynamics. This may cause instability, possibly causing the robot to harm itself and its environment. To prevent this while not placing overly restrictive constraints on the energy generated by the controller, we place a time-varying constraint on the system's power. The power limit varies with a heuristic measure of a desired task trajectory's stability, which is based on the largest Lyapunov exponent. When the trajectory is deemed unstable, the controller is forced to dissipate energy, while it is allowed to generate energy when the trajectory is stable. The constraint is included in a strict task-priority framework, allowing a redundant robotic platform to perform several tasks simultaneously while ensuring that the performance of the higher-priority tasks is not affected by the lower-priority tasks. The presented method is validated by simulation of an articulated intervention autonomous underwater vehicle (AIAUV).

Keywords: Redundant manipulators, Vehicle-manipulator systems, Marine systems, Passive, Hierarchical control, Optimization

1. INTRODUCTION

This paper is motivated by the field of underwater vehicle-manipulator systems (UVMSs). These robots can perform various tasks related to underwater inspection, maintenance and repair (IMR), thus dramatically decreasing the risk to human life compared to having human divers perform the same operations. Today, the majority of underwater IMR tasks are performed by either human divers or remotely operated vehicles (ROVs) supported by large surface vessels. These vessels are currently the main contributor to both the environmental impact and the cost of IMR operations (Liljebäck and Mills, 2017), so by making the UVMSs autonomous, the operations become greener, safer and cheaper.

One particularly interesting group of UVMSs in this regard is articulated intervention autonomous underwater vehicles (AIAUVs), see Fig. 1. AIAUVs are UVMSs with both the hovering and intervention capabilities of classic ROVs, as well as the favorable hydrodynamics of survey AUVs, making them well suited for IMR tasks. In this paper, we will focus on operations that require physical interaction with the environment, such as maintenance and reparation tasks.

When a robot goes in and out of contact with its environment, the dynamics of the controlled system, i.e. the robot, changes. This change in dynamics may lead to instability, which again may cause harm to both the robot and its surroundings. To avoid this, a popular strategy is to design the controller such that the relationship between the generalized error velocities and the generalized control force is *passive*.

Fig. 1. The Eelume AIAUV performing an inspection task.

As the feedback connection of two passive systems is passive, and, under certain observability conditions, asymptotically stable (Khalil, 2002, Thms. 6.1&6.3), designing the controller to render the closed-loop system passive ensures the (asymptotic) stability of the closed-loop system if the environment is also passive. In fact, it can be shown that any non-passive robot can be destabilized by a passive environment (Stramigioli, 2015).

Some tasks, however, may require more control effort than passivity typically allows. To circumvent this problem, one may introduce so-called *virtual energy tanks*. These are virtual reservoirs of pre-allocated energy that can be used to execute tasks that would otherwise be impossible

⋆ This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program, 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.

10.1016/j.ifacol.2024.10.078

to accomplish without violating the passivity constraint. These reservoirs can also be refilled with the energy dissipated by the system without violating passivity. A downside of this approach is that one needs an "energy budget" - an estimate of how much energy is needed to accomplish a task. If the estimate is too low, the task will be interrupted prematurely, whereas a too-large energy budget no longer ensures safety. One solution is to only fill the energy tanks with the least amount of initial energy needed to fulfill the task (Schindlbeck and Haddadin, 2015), but as pointed out in e.g. Benzi et al. (2022), this is insufficient for systems with time-varying environmental dynamics. In Shahriari et al. (2018), *valve-based* virtual energy tanks were introduced to enable constraints on the power, as energy tanks in themselves do not constrain how fast the energy can be used, which may lead to unsafe behavior. The energy tank approach was abandoned in Cuniato et al. (2022), where it was proposed to instead use only a constraint on the produced power. A large benefit of this method is that it removes the need for an energy budget, and the approach proposed in this paper will therefore build on Cuniato et al. (2022) to handle the interaction forces.

Vehicle-manipulator systems (VMSs) are in general redundant with respect to payload pose tasks, as they have a mobile base and a manipulator arm. The control framework therefore needs to handle this redundancy in tandem with the interaction. One way of doing redundancy resolution is through task-priority schemes.

Task-priority schemes are typically based on either null-space projections (Hanafusa et al., 1981; Siciliano and Slotine, 1991) or optimization (Kanoun et al., 2011). The null-space projection-based methods achieve *strict* task priority, meaning that the higher-priority tasks' performances are unaffected by the lower-priority tasks. Usually, however, these schemes cannot incorporate inequality tasks, such as safety-critical tasks like respecting joint limits and avoiding obstacles, though the introduction of activation functions (Simetti et al., 2014, 2018; Cieślak and Ridao, 2018) and tangent cones (Moe et al., 2016) allows the inclusion of scalar inequality tasks. Conversely, optimization-based methods can handle inequality tasks, but typically only achieve *soft* task priority, meaning that the higher-priority tasks' performances are prioritized over the lower-priority tasks, but not necessarily unaffected by them. There are, however, optimization-based frameworks, like Kanoun et al. (2011), that achieve strict task priority as well.

While the null-space projection-based methods presented in Dietrich and Ott (2020); Garofalo and Ott (2020) have recently been extended to (U)VMSs (Sæbø et al., 2022; Dyrhaug et al., 2023), few optimization-based solutions have been explored for these platforms. One optimization-based task-priority framework that has been explored, however, is the one proposed in Basso and Pettersen (2020). This framework allows for strict priority between an arbitrary number of priority levels, and soft priority between the tasks within each level. As the framework is based on optimization, it can also handle inequality tasks. However, no tasks handling the stability issues stemming from interaction with the environment, e.g. a task ensuring passivity, were included in the paper.

Passivity in task-priority schemes has been a topic of interest for the last few years. As null-space projections violate passivity (Dietrich et al., 2016), energy tank-based solutions have been proposed in Dietrich et al. (2016, 2017); Michel et al. (2020, 2022). In Michel et al. (2022), power-based constraints are also introduced. All these methods are, however, still reliant on energy tanks, and thus on energy budgets, and do not allow inequality tasks.

In this paper, we present an approach to handle the redundancy in tandem with the interaction for redundant robot manipulators, like UVMSs. To handle the interaction, a constraint inspired by Cuniato et al. (2022) is placed on the produced power with respect to the top non-safety-critical tasks in the task hierarchy. Contrary to Cuniato et al. (2022), however, the constraint is not placed as a "safety-layer" on top of an existing controller, but is instead incorporated in the task-priority framework proposed in Basso and Pettersen (2020), thus keeping the advantageous strict hierarchy between tasks. The proposed method is validated through a simulation of an AIAUV, and the results are compared to those of the same controller without the power constraint in place.

The paper is organized as follows: In Section 2, the mathematical model of the dynamical system assumed in this paper, along with the specific model for a UVMS, is presented. The task-priority framework, along with some required background material, is presented in Section 3, before the proposed power constraint and how it changes with the observed stability of the system is presented in Section 4. The simulation results validating the proposed method are presented in Section 5, and finally, in Section 6, conclusions and future work are presented.

2. MODEL

In this section, the general model assumed for the control system is presented first, before the specific model for UVMSs, the class of robots that inspired this paper, is presented.

2.1 General model

The control system is assumed to be nonlinear and control affine, i.e. of the form

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

where $x \in D \subset \mathbb{R}^p$ are the controlled states, $u \in U \subset \mathbb{R}^r$ is the control input, and f and g are locally Lipschitz. With

$$y = \sigma(x) - \sigma_d(x) \tag{2}$$

denoting the error of some equality task $\sigma : \mathbb{R}^p \to \mathbb{R}^m$, the input-output dynamics become

$$y^{(\rho)} = \underbrace{L_f^{\rho} y(x)}_{b(x)} + \underbrace{L_g L_f^{\rho-1} y(x)}_{A(x)} u \tag{3}$$

with $\rho \in \mathbb{N}$ the relative degree of the system (1)-(2), if $L_g L_f^k y = 0$ for $0 \leq k \leq \rho - 2$ and $L_g L_f^{\rho-1} y \neq 0$. Note that we will slightly abuse notation throughout this paper, and denote

$$L_g h(x) = \frac{\partial h(x)}{\partial x} g(x)$$

for the sake of compactness.

The system (1) can be decomposed into the external dynamics, with state $\eta = [y^T, \dot{y}^T, \ldots, y^{(\rho-1)T}]^T \in X \subset$

$\mathbb{R}^{\rho m}$ and the internal dynamics, with state $z \in Z \subset \mathbb{R}^{p-\rho m}$:

$$\dot{\eta} = \bar{f}(\eta, z) + \bar{g}(\eta, z)u \tag{4a}$$
$$\dot{z} = f_z(\eta, z). \tag{4b}$$

Here, $\bar{f}(\eta, z) = F\eta + Gb(x)$ and $\bar{g}(\eta, z) = GA(x)$, where

$$F = \begin{bmatrix} 0 & I & 0 & \dots & 0 \\ 0 & 0 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & I \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ I \end{bmatrix}, \tag{5}$$

with 0 being the $m \times m$ zero matrix and I the $m \times m$ identity matrix.

2.2 UVMS model

The dynamic model of a UVMS with n joints (From et al., 2014, Chap. 8.2) is rewritten to fit the control affine form of (1) with $x = [x_1^T, x_2^T]^T$:

$$f(x) = \begin{bmatrix} J_a(x_1)x_2 \\ -M(x_1)^{-1}\left(C(x)x_2 + D(x)x_2 + g(x_1) - \tau_e\right) \end{bmatrix}, \tag{6a}$$

$$g(x) = \begin{bmatrix} 0 \\ M(x_1)^{-1}B(x_1) \end{bmatrix} \tag{6b}$$

where $x_1 = [p^T, q^T, \theta^T]^T \in \mathbb{R}^{7+n}$, with $p \in \mathbb{R}^3$ being the position of the base in the inertial frame, $q \in \mathbb{R}^4$ a unit quaternion describing the orientation of the base and $\theta \in \mathbb{R}^n$ the joint angles, and $x_2 = [v^T, \omega^T, \dot{\theta}^T]^T \in \mathbb{R}^{6+n}$, with $v, \omega \in \mathbb{R}^3$ describing the linear and angular velocity of the base, respectively, and $\dot{\theta}$ the joint angle velocities. Furthermore, M is the inertia matrix, C the Coriolis and centripetal force matrix, D the damping matrix, g the gravitational and buoyancy forces, τ_e the external generalized forces, B the actuator configuration matrix and J_a is the Jacobian transforming body-fixed velocities x_2 to velocities in the inertial frame $\dot{x}_1$.

3. TASK-PRIORITY FRAMEWORK

In this section, we give an introduction to the task-priority framework (Basso and Pettersen, 2020) that we will utilize in this paper, along with some necessary background material.

3.1 Background material

Control Lyapunov functions A control Lyapunov function (CLF) for the control affine system (1) is a positive definite function $V(x)$ that satisfies

$$\inf_{u \in U} [L_f V(x) + L_g V(x)u] < -\beta(V(x)),$$

where β is a class $\mathcal{K}$ function. In Ames et al. (2014), a specific CLF type was introduced to explicitly control the exponential convergence rate through a parameter ϵ:

Definition 1. A continuously differentiable function $V_\epsilon : X \to \mathbb{R}$ is a *rapidly exponentially stabilizing control Lyapunov function (RES-CLF)* for the system (4) if there exist constants $a_1, a_2, a_3 > 0$ such that for all $0 < \epsilon < 1$ and for all $(\eta, z) \in X \times Z$

$$a_1\|\eta\|^2 \le V_\epsilon(\eta) \le \frac{a_2}{\epsilon^2}\|\eta\|^2, \tag{7}$$

$$\inf_{u \in U} [L_{\bar{f}} V_\epsilon(\eta, z) + L_{\bar{g}} V_\epsilon(\eta, z)u] \le -\frac{a_3}{\epsilon}V_\epsilon(\eta). \tag{8}$$

A RES-CLF can be constructed by first solving the continuous-time Riccati equation and then define

$$V_\epsilon(\eta) = \eta^T \begin{bmatrix} \frac{1}{\epsilon}I & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} \frac{1}{\epsilon}I & 0 \\ 0 & I \end{bmatrix} \eta := \eta^T P_\epsilon \eta, \tag{9}$$

where P is the solution of the Riccati equation.

Control barrier functions A control barrier function (CBF) is used to render a set $\mathcal{C} := \{x \in D : h(x) \ge 0\}$ of the state space forward invariant, meaning that for every $x_0 \in \mathcal{C}$, $x(t) \in \mathcal{C}$, with $x_0 = x(t_0)$, and $t \in [t_0, \infty)$. The standard definition of a CBF, however, assumes that the constraint $h(x) \ge 0$ has a relative degree of one, i.e. that $\dot{h}$ depends on the control input u. For robotic systems, this is often not the case, as the set we want to render forward invariant typically is related to the robot pose, while the control input usually is a generalized force coming in at the acceleration level.

In the exponential CBF (ECBF) formalism, however, $h(x)$ may have an arbitrarily high relative degree $r > 1$, meaning

$$h^{(r)}(x, u) = L_f^r h(x) + L_g L_f^{r-1} h(x)u$$

with $L_g L_f^{r-1} h(x) \ne 0$ and $L_g L_f^k h(x) = 0$ for $1 \le k \le r - 2$. Defining $\eta_b(x) := [h^T(x), \dot{h}^T(x), \dots, h^{(r-1)T}(x)]^T$, the dynamics of h can then be written as a linear system, and the ECBF can be defined as follows (Ames et al., 2019):

Definition 2. Given a set $\mathcal{C} \subset D \subset \mathbb{R}^p$ defined as the superlevel set of an r-times continuously differentiable function $h : D \to \mathbb{R}$, the function h is an *exponential control barrier function (ECBF)* if there exists a row vector $K_\alpha \in \mathbb{R}^r$ such that for the control affine system (1) and $\forall x \in \mathrm{Int}(\mathcal{C})$

$$\sup_{u \in U} [L_f^r h(x) + L_g L_f^{r-1} h(x)u] \ge -K_\alpha \eta_b(x). \tag{10}$$

Combining CLFs and ECBFs As both the RES-CLF (8) and the ECBF conditions (10) are affine with respect to the control input u, the control problem can be formulated as a quadratic program (QP) with the conditions as constraints (Ames et al., 2019):

$$\begin{aligned} \min_{u \in U, \delta \in \mathbb{R}} \quad & \frac{1}{2}u^T H(x)u + c^T(x)u + w\delta^2 \\ \text{subject to} \quad & \\ & L_{\bar{f}} V_\epsilon(\eta, z) + L_{\bar{g}} V_\epsilon(\eta, z)u \le -\frac{a_3}{\epsilon}V_\epsilon(x) + \delta \\ & L_f^r h(x) + L_g L_f^{r-1} h(x)u \ge -K_\alpha \eta_b(x). \end{aligned} \tag{11}$$

Here, $H : D \to \mathbb{R}^{m \times m}$ is any positive semi-definite matrix, $c : D \to \mathbb{R}^m$, and $\delta \in \mathbb{R}$ is a slack variable penalized by $w > 0$. Note that a slack variable has only been added to the CLF-based constraint, effectively creating a strict prioritization of the CBF-based constraint over the CLF-based constraint.

3.2 Task-priority framework

The framework presented in Basso and Pettersen (2020) ensures strict task priority through iteratively solving QPs, where the QPs corresponding to the highest prioritized task levels are solved first, before the QPs for the lower level tasks are solved subsequently. The higher priority tasks are implemented as strong (slack parameter-free) constraints in the lower priority tasks' QPs, while the tasks at that priority level are implemented as soft constraints (with slack parameters), mirroring the prioritization of the CBF inequality over the CLF inequality in (11).

Soft priority between the tasks at each task level is implemented by relating different-valued penalty parameters w to the slack variables δ corresponding to each task.

The QP in (11) can be extended to include several equality and inequality tasks. The input-output dynamics for each of the N equality tasks are given by

$$y_i^{(\rho_i)} = L_f^{\rho_i} y_i(x) + L_g L_f^{\rho_i - 1} y_i(x) u \qquad i = 1, \ldots, N, \tag{12}$$

and the external dynamics states η_i and RES-CLFs $V_{\epsilon,i}$ are defined analogously to (4a), (5) and (9). Furthermore, with M inequality tasks described by the superlevel sets $\mathcal{C}_j$ of r_j times continuously differentiable functions $h_j(x)$, the control input can be obtained by solving the following QP:

$$\begin{aligned} &\min_{u \in U, \delta \in \mathbb{R}^N} \quad u^T H(x) u + c^T(x) + \delta^T W \delta \\ &\text{subject to} \\ &L_{\bar{f}_i} V_{\epsilon,i} + L_{\bar{g}_i} V_{\epsilon,i} u \leq -\frac{a_{3,i}}{\epsilon} V_{\epsilon,i} + \delta_i \quad i = 1, \ldots, N \\ &L_f^{r_j} h_j + L_g^{r_j - 1} h_j u \geq -K_{\alpha,j} \eta_{b,j} \quad j = 1, \ldots, M \end{aligned} \tag{13}$$

where $W \in \mathbb{R}^{m \times m}$ is the diagonal matrix of penalty parameters and η_b are defined similarly as in Section 3.1.2.

To establish more than two priority levels, a new QP is solved for every level below the first two (which can be established by a single QP as in (13)). The control input u_1 obtained from the first QP can be refined to solve tasks at lower priority levels without affecting the $N_0 + N_1$ equality tasks of the first two priority levels by enforcing $L_{\bar{f}_i} V_{\epsilon,i} + L_{\bar{g}_i} V_{\epsilon,i} u \leq L_{\bar{f}_i} V_{\epsilon,i} + L_{\bar{g}_i} V_{\epsilon,i} u_1$, or equivalently

$$L_{\bar{g}_i} V_{\epsilon,i} u \leq L_{\bar{g}_i} V_{\epsilon,i} u_1 \tag{14}$$

for $i = 1, \ldots, N_0 + N_1$. Likewise, the $M_0 + M_1$ inequality tasks of the first two priority levels can be left unaffected by enforcing

$$L_g L_f^{r_j - 1} h_j u \geq L_g L_f^{r_j - 1} h_j u_1. \tag{15}$$

For an arbitrary priority level p, the control input u_p can then be obtained by solving the following QP:

$$\begin{aligned} &\min_{u_p \in U, \delta \in \mathbb{R}^{N_p}, s \in \mathbb{R}^{M_p}} \quad u_p^T H u_p + c^T u_p + \delta^T W_p \delta + s^T K_p s \\ &\text{subject to} \\ &L_{\bar{g}_i} V_{\epsilon,i} u_p \leq L_{\bar{g}_i} V_{\epsilon,i} u_{p-1} \quad i = 1, \ldots, \bar{N}_{p-1} \\ &L_g L_f^{r_j - 1} h_j u_p \geq L_g L_f^{r_j - 1} h_j u_{p-1} \quad j = 1, \ldots, \bar{M}_{p-1} \\ &L_{\bar{f}_k} V_{\epsilon,k} + L_{\bar{g}_k} V_{\epsilon,k} u_p \leq -\frac{a_{3,k}}{\epsilon} V_{\epsilon,k} + \delta_k \quad k = \bar{N}_{p-1} + 1, \ldots, \bar{N}_p \\ &L_f^{r_l} h_l + L_g^{r_l - 1} h_l u_p \geq -K_{\alpha,l} \eta_{b,l} - s_l \quad l = \bar{M}_{p-1} + 1, \ldots, \bar{M}_p. \end{aligned} \tag{16}$$

Here, the slack variables s are penalized by elements in the diagonal matrix $K_p \in \mathbb{R}^{M_p \times M_p}$, $\bar{N}_p = N_0 + N_1 + \cdots + N_p$ and $\bar{M}_p = M_0 + M_1 + \cdots + M_p$, where N_p and M_p are the number of equality and inequality tasks at priority level p, respectively.

4. POWER CONSTRAINT

In this section, we propose a constraint on the produced power to enhance the safety of the system without the need for an energy budget. The power constraint is motivated by Cuniato et al. (2022), and we adapt this to use a heuristic stability measure instead of the largest Lyapunov exponent (LLE) used in their work. Based on this power constraint, we propose an expression for the power bound that, contrary to that in Cuniato et al. (2022), allows different gains for when the desired trajectory is found to be stable and when it is not. Lastly, we show how this power constraint can be incorporated in the task-priority framework presented in Section 3 to achieve enhanced safety without sacrificing the strict task priority.

4.1 Power constraint

The produced power with respect to a task of size m is given by the inner product of the control wrench $\tau_c \in \mathbb{R}^m$ and the generalized velocity error $\tilde{\nu} \in \mathbb{R}^m = \nu - \nu_d$ related to that task. Here, ν and ν_d are the generalized velocities and the desired generalized velocities, respectively. The proposed upper bound on the produced power can thus be written as

$$\tilde{\nu}^T \tau_c \leq \bar{p}(\lambda^*, \tilde{\nu}). \tag{17}$$

Notice that $\bar{p}$ is not constant, but varies with $\tilde{\nu}$ and λ^*, the heuristic stability measure related to the task. The dependency on λ^* rather than the LLE is the sole difference between (17) and the constraint proposed in Cuniato et al. (2022). The computation of this stability measure will now be presented.

4.2 Stability measure

The stability measure is inspired by, but not necessarily equal to, the LLE of the task trajectory. The Lyapunov exponents of a dynamical system define the exponential rate of convergence or divergence of trajectories starting close to each other. If the system has a positive LLE, it indicates that the trajectories are unstable, while a negative LLE indicates the asymptotic stability of the trajectory in question.

To compute the stability measure, we use the method proposed in Dabrowski (2012) for computing the finite-time LLE:

$$\lambda^* = \frac{\tilde{x}^T \dot{\tilde{x}}}{\tilde{x}^T \tilde{x}}, \tag{18}$$

where λ^* is the LLE and $\tilde{x}(t) = x_{nom}(t) - x_p(t)$ is the perturbation vector between a nominal solution $x_{nom}(t)$ and a perturbed solution $x_p(t)$ of a dynamical system $\dot{x} = F(x)$. In our case, $\tilde{x}$ is given by $\tilde{x} = x - x_d$, and λ^* thus becomes a measure of the rate at which $\tilde{x}$ is instantaneously converging to or diverging from zero.

The solution λ^* of (18) is, however, not necessarily the LLE of the system in our case. The equation yields the biggest real part of the eigenvalues of the Jacobian $\frac{dF}{dx}(x(t))$, but the differentiability of the system $\dot{x} = F(x)$ cannot be guaranteed in general, which is a prerequisite for λ^* to be an LLE. Specifically, the QPs used in the task-priority framework presented in Section 3 may give a solution that is not differentiable in x. The stability measure given by (18) is thus not necessarily an LLE, but still provides a measure of the instantaneous convergence of the system to the desired trajectory. It can therefore be used as a heuristic stability measure. In practice, low-pass filtering of λ^* is advised to avoid e.g. noise-related problems.

4.3 Power bound

To enhance safety, we want to dissipate energy when the state x diverges from the desired state x_d ($\lambda^* > 0$). Simultaneously, we want to allow the system to generate energy when x is converging to x_d ($\lambda^* < 0$) to improve the performance of the task. We do this by slightly altering the expression for the power bound $\bar{p}$, see (17), that was proposed in Cuniato et al. (2022):

$$\bar{p}(\lambda^*, \tilde{\nu}) = \begin{cases} -k_{\lambda,g}\lambda^* & \text{if } \lambda^* \leq 0 \\ -k_{\lambda,d}\lambda^*\tilde{\nu}^T\tilde{\nu} & \text{if } \lambda^* > 0. \end{cases} \quad (19)$$

The alteration lies in the added possibility of choosing different gains $k_{\lambda,g}, k_{\lambda,d} > 0$ for the stable and the unstable case, yielding increased design freedom. This can for instance be used to enforce more aggressive energy dissipation when the system is unstable, while simultaneously preventing too much energy generation from being allowed when the system is stable. Note also the inclusion of $\tilde{\nu}$ in the second case. As in Cuniato et al. (2022), this is added to ensure the feasibility of the energy dissipation constraint, as the system cannot dissipate energy if there is no (kinetic) energy to dissipate.

4.4 Inclusion in the task-priority framework

In Cuniato et al. (2022), the power constraint is implemented as a CBF, either placing constraints on the state x in case of a state feedback controller $u = k(x)$, or directly on the control input u through an integral CBF (I-CBF) (Ames et al., 2021). When the I-CBF is used, the state is augmented with the control input, and the optimization problem is solved for the time derivative of the control input, $\dot{u}$. Neither of these solutions are fitting for the task-priority framework we adopt in this work.

Firstly, as we do not have a closed-form solution to the optimization problem yielding u, placing a constraint on u through a constraint on x is not appropriate. Secondly, using an I-CBF, one could either place it inside the task-priority framework by having all the optimization problems solve for $\dot{u}$ instead of u, or place the I-CBF as a "layer" on top of the existing controller, adapting the control input in a minimally invasive fashion to comply with the power constraint. Neither of these methods would be a good fit, though, as the first would entail time differentiating the entries of the dynamic model used in the CLF- and CBF-based tasks of the hierarchy, while the second would destroy the strict task priority. Instead, the power constraint (17) is implemented as-is, abandoning the CBF paradigm.

The constraint should not be placed at the top priority level, as the satisfaction of e.g. joint and actuator constraints is more important, but should ideally be placed above pure performance tasks such as the tracking of a desired trajectory.

Strict priority can be achieved by enforcing the power constraint not only on the solution of the QP corresponding to the priority level where the power constraint task is placed, but on the solution of every QP corresponding to a priority level below that one. In case tasks from a level prioritized higher than the one where the power constraint task is introduced require the power constraint to be violated, the solutions of the QPs corresponding to priority levels below the power constraint task are then constrained to not violate the power constraint (17) any more than was required by the higher-priority tasks. Specifically, this is done by including the constraint

$$\tilde{\nu}^T\tau_{c,p} \leq \bar{p} + \delta^*_{\bar{p}} \quad (20)$$

in all those lower-level QPs, following the lines of Kanoun et al. (2011). Here, $\tau_{c,p}$ is the control wrench yielded by the QP to be solved for the current priority level, while $\delta^*_{\bar{p}}$ is a constant whose value is equal to that of the slack variable associated with the power constraint task. We will now present an example of what a task hierarchy including a power constraint could look like for a UVMS with dynamics given by (6).

Task hierarchy example In this example, three set-based and four equality tasks, distributed on four priority levels, are incorporated in the task-priority framework as given in Table 1.

Table 1. The task hierarchy.

Level	**Tasks**
1	Joint angle limits and actuator limits
2	Power constraint
3	Base orientation, base y- and z-position, and base force in x-direction
4	Joint angle velocities

The power constraint (17) is implemented at the second priority level, and is implemented as individual constraints for each of the base's degrees of freedom (DoFs):

$$[c_1, \ldots, c_6]^T := \text{diag}(\tilde{\nu})[I_6, 0_{6\times n}]Bu \leq [\bar{p}_1, \ldots, \bar{p}_6]^T. \quad (21)$$

This allows a soft prioritization between the power constraints along the different DoFs. Here, the velocity error $\tilde{\nu}$ is set as $\tilde{\nu} = [v^T - v_d^T, \omega^T - \omega_d^T]^T$, where v and ω are as in Section 2.2, and v_d and ω_d are the desired linear and angular velocities of the base, respectively. The desired velocity along the inertial x-axis is defined as zero.

The base force- and pose-related tasks are placed at the third priority level. To control the force the base is to exert along the inertial x-axis, an equality force task is introduced through two inequality constraints, $c_7 := [1, 0_{1\times(5+n)}]Bu \leq f_d$ and $c_8 := [1, 0_{1\times(5+n)}]Bu \geq f_d$, bounding the force from above and below. To fully control the base pose, two CLF-based tasks, a base position task, $y_1 = p^i_{b_{yz}} - p^i_{b_{yz},d}$, concerning the position along the inertial y- and z-axis, and a base orientation task, $y_2 = \tilde{\varepsilon}$, are also included at the same priority level. Here, f_d is the desired force, $p^i_{b_{yz}}$ and $p^i_{b_{yz},d}$ are the base's actual and desired position along the inertial y- and z-axes, respectively, and $\tilde{\varepsilon}$ is the imaginary part of the error quaternion $\tilde{q} = q_d^* \otimes q$, where q and q_d are unit quaternions representing the actual and the desired base orientation, respectively. Note that the force task is neither CBF- nor CLF-based, and is therefore incorporated in the task-priority framework similarly to the power constraint task, cf. (20).

The most safety-critical tasks are introduced at the top priority level, which in this case are constraints on the actuators' minimum and maximum output, along with $2n$ ECBFs to avoid the joint angle limits, $h_i = \theta_i - \theta_{i,\min}$ and $h_{i+n} = \theta_{i,\max} - \theta_i$ for $i = 1, \ldots, n$. At the lowest priority level, CLF-based equality tasks for the joint velocities, $y_{2+j} = \dot{\theta}_j - \dot{\theta}_{j,d}$ for $j = 1, \ldots, n$, are introduced to fully define the desired behavior of the robot.

As the power is constrained individually for each DoF in (21), the stability measure λ^* is also computed individually for each DoF. This, again, means that only individual errors $\tilde{x}$ and their derivatives $\dot{\tilde{x}}$ are used when calculating λ^* through (18). Thus, we avoid adding quantities with different units, which would have been the case if we had used e.g. both the position error and the orientation error in the same computation. The errors, $\tilde{x}$ and $\dot{\tilde{x}}$, are set as the position error and its derivative for the position task, the vector part of the quaternion error and its derivative for the orientation task, and the integral of the force error and the force error itself for the power constraint task. As in Cuniato et al. (2022), using the force error and its integral is preferred over using the force error and its time derivative, as the latter is noisy in practice.

5. SIMULATIONS

To showcase the merit of the proposed method, the control scheme was implemented on a model of an AIAUV (see Fig. 1) interacting with a spring that was suddenly removed after 2 s, and compared with the same control scheme, but without the power constraint. The simulation was done in Matlab/Simulink, using an ode3 solver with a fixed time step of 0.001 s.

The task hierarchy and the tasks therein were set as described in Section 4.4.1, with the gains and the weights for the penalty parameters of the non-CBF-based tasks as given in Table 2. The power constraint gains for dissipation, $k_{\lambda,d}$, were set to 1000, 10 and 1 for the base's linear velocity along the inertial x-axis, the y- and the z-axis, and for the angular velocity about each axis, respectively, while the gains for the generation, $k_{\lambda,g}$ were set as $k_{\lambda,d}/10^5$ to better match the slow dissipation caused by the low velocities in our case. The desired force was set to $f_d = 10$ N, while the desired joint angle velocities were set to zero. The desired base pose, not including the position along the inertial x-axis, was set to the initial one.

Table 2. The penalty parameters w and convergence rates ϵ used in the simulations.

	$c_{1:6}$	$c_{7:8}$	y_1	y_2	$y_{3:2+n}$
w	1000	10	60	60	10
ϵ	-	-	0.9	0.9	0.9

As in Basso and Pettersen (2020), the objective functions for the QPs described by (16) were set with $H = A^T A$ and $c = 2A^T b$, where $A = [A_1^T A_2^T A_3^T]^T$ and $b = [b_1^T b_2^T b_3^T]^T$ are the vertical concatenations of the A- and b-matrices from (3) for the three CLF-based tasks in the simulation.

5.1 Simulation results

The results of the simulations are summarized in Figs. 2 and 3. In Fig. 2, we see the power generated by the controller with respect to the base along the inertial x-axis with and without the power constraint in place, while Fig. 3 shows the position of the base along the inertial x-axis in the two cases. It is clear from Fig. 2 that the power constraint is only violated very slightly when it is applied, and that the power constraint is adapted such that once the spring is removed, and the force task diverges from its desired value, the constraint forces the controller to dissipate energy. From Fig. 3 we can see that this dissipation stops the base from moving too far once the spring is removed, only allowing the base to move about 0.1 mm in 18 s. Comparatively, the base moves more than 4 m in 18 s when the power constraint is not in place.

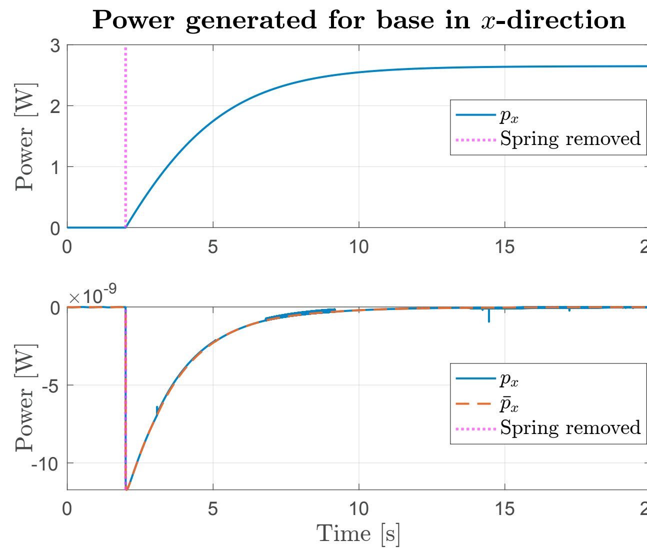

Fig. 2. The power p_x generated by the controller with respect to the base along the inertial x-direction with (upper plot) and without (lower plot) the power constrained to be less than $\bar{p}_x$.

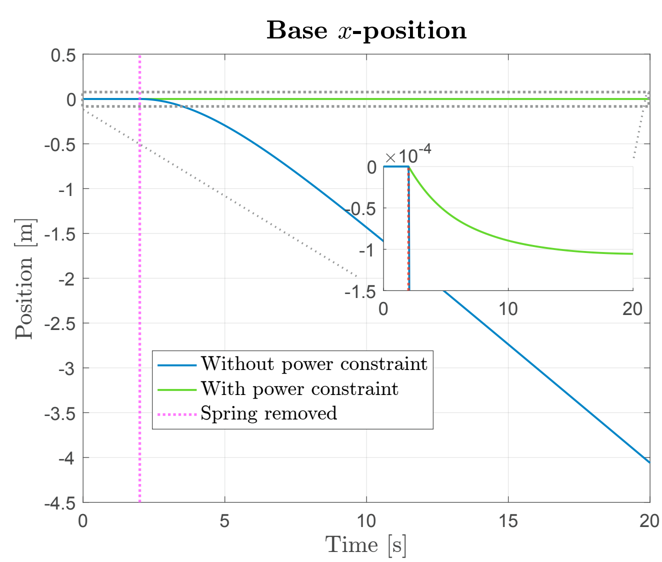

Fig. 3. The base position along the inertial x-axis with and without the power constraint in place.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a control method for handling the redundancy of robot manipulators, like UVMSs, in tandem with handling the physical interaction of the manipulator with its environment. Specifically, we have introduced a constraint on the produced power with respect to a task to increase the safety of a redundant robotic manipulator. The constraint is implemented in a way that respects the strict priority of the task-priority framework, and also varies with the instantaneous convergence of the task trajectory, enabling power to be generated when the state trajectory is converging to the desired one, and enforcing power dissipation when it is not. The proposed method was validated through a simulation of an AIAUV, where the merit of the method was demonstrated by comparing the results to those of the same controller without the power constraint in place. In future work, the method will be validated experimentally.

REFERENCES

Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. (2019). Control Barrier Functions: Theory and Applications. In *Proc. 18th European Control Conf.*

Ames, A.D., Galloway, K., Sreenath, K., and Grizzle, J.W. (2014). Rapidly Exponentially Stabilizing Control Lyapunov Functions and Hybrid Zero Dynamics. *IEEE Trans. Automat. Contr.*, 59(4).

Ames, A.D., Notomista, G., Wardi, Y., and Egerstedt, M. (2021). Integral Control Barrier Functions for Dynamically Defined Control Laws. *IEEE Control Syst. Lett.*, 5(3).

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 Congress.*

Benzi, F., Brunner, M., Tognon, M., Secchi, C., and Siegwart, R. (2022). Adaptive Tank-based Control for Aerial Physical Interaction with Uncertain Dynamic Environments Using Energy-Task Estimation. *IEEE Robot. Autom. Lett.*, 7(4).

Cieślak, P. and Ridao, P. (2018). Adaptive Admittance Control in Task-Priority Framework for Contact Force Control in Autonomous Underwater Floating Manipulation. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.*

Cuniato, E., Lawrance, N., Tognon, M., and Siegwart, R. (2022). Power-Based Safety Layer for Aerial Vehicles in Physical Interaction Using Lyapunov Exponents. *IEEE Robot. Autom. Lett.*, 7(3).

Dabrowski, A. (2012). Estimation of the largest Lyapunov exponent from the perturbation vector and its derivative dot product. *Nonlinear Dynamics*, 67(1).

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

Dietrich, A., Ott, C., and Stramigioli, S. (2016). Passivation of Projection-Based Null Space Compliance Control Via Energy Tanks. *IEEE Robot. Autom. Lett.*, 1(1).

Dietrich, A., Wu, X., Bussmann, K., Ott, C., Albu-Schäffer, A., and Stramigioli, S. (2017). Passive Hierarchical Impedance Control Via Energy Tanks. *IEEE Robot. Autom. Lett.*, 2(2).

Dyrhaug, J.I., Tveter, E., Schmidt-Didlaukies, H.M., Basso, E.A., Pettersen, K.Y., and Gravdahl, J.T. (2023). Robust Hierarchical Tracking Control of Vehicle-Manipulator Systems. In *Proc. IFAC World Congress.*

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.

Garofalo, G. and Ott, C. (2020). Hierarchical Tracking Control With Arbitrary Task Dimensions: Application to Trajectory Tracking on Submanifolds. *IEEE Robot. Autom. Lett.*, 5(4).

Hanafusa, H., Yoshikawa, T., and Nakamura, Y. (1981). Analysis and Control of Articulated Robot Arms with Redundancy. *IFAC Proceedings Volumes*, 14(2).

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

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.*

Michel, Y., Ott, C., and Lee, D. (2020). Passivity-based variable impedance control for redundant manipulators. *IFAC-PapersOnLine* In *Proc. IFAC World Congress*, 53(2).

Michel, Y., Ott, C., and Lee, D. (2022). Safety-Aware Hierarchical Passivity-Based Variable Compliance Control for Redundant Manipulators. *IEEE Trans. Robot.*, 38(6).

Moe, S., Antonelli, G., Teel, A.R., Pettersen, K.Y., and Schrimpf, J. (2016). Set-Based Tasks within the Singularity-Robust Multiple Task-Priority Inverse Kinematics Framework: General Formulation, Stability Analysis, and Experimental Results. *Front. Robot. AI*, 3.

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

Schindlbeck, C. and Haddadin, S. (2015). Unified passivity-based Cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks. In *Proc. IEEE Int. Conf. Robot. Autom.*

Shahriari, E., Johannsmeier, L., and Haddadin, S. (2018). Valve-based Virtual Energy Tanks: A Framework to Simultaneously Passify Controls and Embed Control Objectives. In *Proc. American Control Conf.*

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.*

Simetti, E., Casalino, G., Wanderlingh, F., and Aicardi, M. (2018). Task priority control of underwater intervention systems: Theory and applications. *Ocean Engineering*, 164.

Simetti, E., Casalino, G., Torelli, S., Sperindé, A., and Turetta, A. (2014). Floating Underwater Manipulation: Developed Control Methodology and Experimental Validation within the TRIDENT Project: Floating Underwater Manipulation. *Journal of Field Robotics*, 31(3).

Stramigioli, S. (2015). Energy-Aware Robotics. In M.K. Camlibel, A.A. Julius, R. Pasumarthy, and J.M. Scherpen (eds.), *Mathematical Control Theory I*, Lecture Notes in Control and Information Sciences. Springer International Publishing.