Check for updates

Received: 29 March 2023 | Revised: 1 July 2023 | Accepted: 13 September 2023

DOI: 10.1049/cth2.12558

IET Control Theory & Applications

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Learning passive policies with virtual energy tanks in robotics

Riccardo Zanella[1] | **Gianluca Palli**[1] | **Stefano Stramigioli**[2] | **Federico Califano**[2]

[1]Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna, Italy

[2]Robotics & Mechatronics (RaM) group, University of Twente, Enschede, The Netherlands

Correspondence
Riccardo Zanella, Department of Electrical, Electronic and Information Engineering, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy.
Email: r.zanella@unibo.it

Funding information
European Commission, Grant/Award Number: 101070136

Abstract

Within a robotic context, the techniques of passivity-based control and reinforcement learning are merged with the goal of eliminating some of their reciprocal weaknesses, as well as inducing novel promising features in the resulting framework. The contribution is framed in a scenario where passivity-based control is implemented by means of virtual energy tanks, a control technique developed to achieve closed-loop passivity for any arbitrary control input. Albeit the latter result is heavily used, it is discussed why its practical application at its current stage remains rather limited, which makes contact with the highly debated claim that passivity-based techniques are associated with a loss of performance. The use of reinforcement learning allows to learn a control policy that can be passivized using the energy tank architecture, combining the versatility of learning approaches and the system theoretic properties which can be inferred due to the energy tanks. Simulations show the validity of the approach, as well as novel interesting research directions in energy-aware robotics.

1 | INTRODUCTION

Robotics is increasingly focusing on the development of control frameworks allowing the transition from industrial cages to unstructured environments. This transition carries the ambitious objective of stable and safe execution of complex tasks where robots coexist with other robots or possibly humans. The practical impossibility of careful dynamic modelling of the environment along those tasks makes the stability objective even more challenging, and model-based approaches poorly suitable.

Passive controllers have been presented as a feasible solution to tackle this problem as the stability of the closed-loop system is in principle independent on the external environmental interaction [1, 2]. Passive systems route physical energy rather than producing it, which is the ultimate reason why stability proofs (using energy functions as Lyapunov candidates) are conveniently assessed in this framework [3, 4]. A powerful technique allowing the passivization of any control action is represented by virtual energy tanks [5–7]. These are able to store information about physical energy flows undergoing the system, and introduce at a control level the scalar information representing the energy budget for the controlled robot. A conditional check on this budget with proper passive reaction strategies in case of depletion of the latter is what guarantees closed-loop passivity. The main drawback of passivity-based control methods (comprising those involving energy tanks) is the lack of optimization over a task performance metric along the design of the controller [8]. This fact led to the claim that passivity-based control methods are associated with a loss of performance, which becomes more severe as the complexity of tasks increases.

On the other side of the control theoretic spectrum, recent advancements in the machine learning community are leading to robots with an outstanding awareness of complex environments and tasks. The intelligence encoded in the control policies, normally optimized by means of performance-based metrics, reflects complex high-level decision-making strategies which are learned thanks to the availability of huge datasets [9–11]. The drawback of these families of approaches is the difficulty in guaranteeing system theoretic properties such as passivity and stability for the controlled system [12].

In this work we merge the passivity-based control (PBC) design based on energy tanks, and reinforcement learning framework (RL), combining system theoretic properties of the closed-loop system induced by the ultimately passive design,

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2024 The Authors. *IET Control Theory & Applications* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

and high-performance achievement peculiar of the RL framework. The proposed idea, presented both in inference and in training, allows to apply PBC to complex robotic tasks, i.e. it introduces a constructive way to passivize controllers which resemble complicated decision-making strategies, like those implemented by RL agents.

1.1 | Related work

We recognize related work casting the energy tank algorithm (seen as a task-agnostic passivization tool) into a framework using task-based information for performance augmentation of the underlying passive system. In ref. [13] the authors introduce the so-called valve-based energy tanks (also used in e.g. ref. [14]), in which extra parameters are introduced on the energy tanks in order to embed control objectives in the design, beyond achieving passivity. Task specifications are translated into tank design rules by controlling the power flows undergoing the system. The idea of using a reference power trajectory in combination with energy tanks is present also in refs. [15, 16], where task-based specifications are used to tune the tank parameters. These works present very stiff task specifications and badly scale to complex tasks in which high-level policies need to be learned. A family of methods using energy tanks and sharing a similar motivation of this work is refs. [17, 18], where an explicit optimization problem is introduced to find the closest passive approximation of a given control action. The idea is to exploit the versatility of energy tank architecture to perform an optimization that can be generalized and scaled for different tasks and whose outcome is a passively controlled system. An important difference between this work and the cited ones (beyond the specific technique to solve the optimization) is that in refs. [17, 18] the desired control action is already given, and the optimization aims at finding the closest passive approximation to it, imposed by a non-depletion constraint of the tank. We will comment on the difference between the approaches, which involves the degree of freedom represented by the initial state in the tank.

Eventually we recognise other contributions relating PBC and learning schemes in general. In ref. [19] the authors use neural ODEs to parametrize and optimize a specific instance of PBC law which possesses an energetic interpretation and generalises a PD controller. In refs. [20, 21] the same PBC law is tuned using an RL scheme. As core difference with respect to this work, the cited ones deal with very specific PBC laws, and as such poorly scale to complex robotic tasks requiring decision-making strategies. The proposed strategy exploits the fact that energy tanks allow for passivization of any controller requiring finite energy, including model free RL policies. In general, these are able to deal with very complex tasks that mimic decision-making strategies, and are far more general than the PBC control laws which can be physically interpreted like those treated in refs. [19–21].

The paper is organized as follows. In Section 2 a throughout explanation of the PBC problem in robotics and its achievement using energy tanks is reviewed, followed by a discussion on the limitations of the methodology. In Section 3 the proposed scheme is presented and validated in Section 4 by means of different simulations. Conclusions and future works are sketched in Section 5.

2 | BACKGROUND AND MOTIVATION

We review, restricted to robotics, the motivation underlying PBC, and how energy tanks formally solve the problem of passivizing an arbitrary control input. We then discuss the limits of the approach, which are tackled with the subsequent RL integration.

2.1 | Passivity-based control and energy tanks

Consider the dynamic model of an n-DoF robot in Lagrangian joint-space coordinates:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + D(q)\dot{q} + \frac{\partial V(q)}{\partial q} = \tau \tag{1}$$

where $q \in \mathbb{R}^n$ are the joint coordinates, $M(q) = M^\top(q) > 0$ is the inertia tensor, $C(q,\dot{q})$ is the matrix collecting centrifugal and Coriolis terms, $D(q) = D^\top(q) \geq 0$ is the matrix collecting friction coefficients, $V : \mathbb{R}^n \to \mathbb{R}$ maps the joint coordinates into the total conservative potentials (e.g. gravity and elastic springs), and $\tau \in \mathbb{R}^n$ are the generalized forces at the joints, collocated with the joint coordinates q. Using the skew symmetric property of the matrix $\dot{M} - 2C$, one obtains that the time derivative of total mechanical energy $E(q,\dot{q}) = \frac{1}{2}\dot{q}^\top M(q)\dot{q} + V(q)$ verifies the inequality

$$\dot{E} = \dot{q}^\top \tau - \dot{q}^\top D(q)\dot{q} \leq \dot{q}^\top \tau \tag{2}$$

which is a statement of passivity for system (1) with the energy function $E(q,\dot{q})$ as *storage function*. Inequality (2) states that system (1) is passive with respect to its energy function $E(q,\dot{q})$, and its so-called *power port* $(\tau,\dot{q})$, an input-output pair whose pairing $\dot{q}^\top\tau$ produces the scalar power flow undergoing the system. Once (2) is integrated in time, the left-hand side of equation $E(t) - E(0)$ represents the energy stored in the system, which is always less than or equal to the energy supplied through the power port, represented by the right term $\int_0^t \dot{q}(s)^\top \tau(s)\mathrm{d}s$. The positive dissipated energy $\int_0^t \dot{q}(s)^\top D(q(s))\dot{q}(s)\mathrm{d}s$ determines the convergence rate to the equilibrium, and acts as a passivity margin, i.e. the higher it is, the bigger is the margin for which the passivity inequality (2) results satisfied[1]. If applied to an (open-loop) physical system like (1), passivity just represents the thermodynamic statement of energy conservation, while if applied to a controlled system, passivity implies stability for the minimum of the storage function under weak conditions qualifying the latter as a Lyapunov function for the equilibrium

[1] If $D = 0$, Equation (2) holds with equal sign and the system is denoted as lossless.

17518652, 0, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/cth2.12558 by University Of Twente FEZ, Wiley Online Library on [07/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

[2, 4]. Next, we illustrate the full motivation behind the need to achieve a passive closed-loop system.

2.1.1 | Passivity as must

To understand the motivation behind a passive design consider system (1) as an *open* one, i.e. a system that, independently on the way it is controlled, can interact with a new, dynamically unknown system, that we call the *s-environment*[2]. This is shown by considering system (1) in which we distinguish the torque contributions in two terms, i.e.

$$\tau = \tau_c + \tau_e \tag{3}$$

where τ_c are the control torques that can be directly applied at the joints by means of the actuators, while $\tau_e = J^\top(q)F$ corresponds to the torques at the joints produced by an external interaction at the end-effector of the robot. Here $F \in \mathbb{R}^6$ is a vector collecting forces and torques at the end-effector in the workspace and $J(q)$ is the Jacobian matrix of the robot. The latter relates joint-space and work-space velocities and forces as the dual relations $\dot{x} = J(q)\dot{q}$ and $\tau_e = J^\top(q)F$, where $\dot{x} \in \mathbb{R}^6$ is the rate of work-space coordinates x, collocated with the forces F, resulting in $\tau_e^\top \dot{q} = F^\top \dot{x}$. Specializing the power balance (2) to the uncontrolled system (1) with the split (3) one obtains

$$\dot{E} = \dot{q}^\top \tau_c + \dot{x}^\top F - \dot{q}^\top D(q)\dot{q} \leq \dot{q}^\top \tau_c + \dot{x}^\top F, \tag{4}$$

i.e. the robotic system (1) is passive with respect to two different power ports: $(\tau_c, \dot{q})$, whose pairing is the power flow undergoing the actuators, and $(F, \dot{x})$, whose pairing is the power flow undergoing the s-environment, an external dynamical system with its own (unknown) dynamics. In this context, the PBC objective can be described as follows

PBC objective. *Given the system (1) with the split (3) inducing power balance (4), design a control law on the port $(\tau_c, \dot{q})$, possibly by designing a controller as a dynamical system, with state $x_c \in \mathbb{R}^c$ and energy function $V_c : \mathbb{R}^c \to \mathbb{R}$, such that for the closed-loop storage function $\mathcal{E}(q, \dot{q}, x_c) = E(q, \dot{q}) + V_c(x_c)$, the passivity condition*

$$\dot{\mathcal{E}} \leq \dot{x}^\top F \tag{5}$$

is verified, i.e. that the controlled system is passive with respect to the s-environment port $(F, \dot{x})$.

This objective is considered of utmost importance (from which the *passivity as must* claim [1]) since, when an interaction takes place ($F \neq 0$) the dynamics of the closed-loop system deeply depends on the dynamics of the s-environment, which cannot be modelled in an oversimplified way, e.g. assuming linearity. This claim is particularly valid if the task to be executed takes place in an unstructured, hazardous scenario where sources of disturbances like unforeseen interactions with humans are possible.

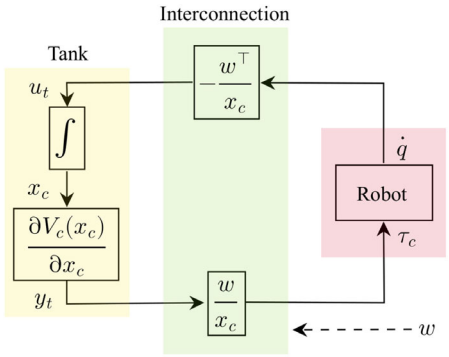


FIGURE 1 The essential of the energy tank algorithm. The external signal w modulates the interconnection (8), here in light green.

The passive design (5) aims at excluding the possibility that in the moment the robot interacts with a passive s-environment, an unbounded amount of energy can be produced during the interaction.[3] This property is referred to as contact stability. For completeness, we report that passivity induces also a stronger property than contact stability, which is sometimes referred to as robust stability [17]: even if the s-environment is not a passive system, contact stability can be assured if the passivity margin of the controlled system covers the energy generated by the non-passive s-environment. All these properties follow from the fact that passive systems are closed under power-preserving interconnection [2, 3], and we refer to ref. [2] for further relations between passivity and different system theoretic stability properties. To conclude, in this work we are aligned with the necessity of achieving the PBC objective when a robotic task needs to be performed.

2.1.2 | Energy tanks

An extremely effective method to achieve the PBC objective relies on the energy tank control algorithm, which we describe next and whose basic schematics is represented in Figure 1.

Mathematically an energy tank is a dynamical system that constitutes an atomic energy-storing element. It can be represented as the (lossless) system:

$$\dot{x}_c = u_t \tag{6}$$

$$y_t = \frac{\partial V_c(x_c)}{\partial x_c}, \tag{7}$$

where $x_c \in \mathbb{R}$ and the energy function is simply $V_c(x_c) = \frac{1}{2}x_c^2$, and as a consequence it presents a power port (u_t, y_t) since $\dot{V}_c = y_t u_t$. The tank (6–7) is interconnected to Equation (1) in a way that allows the implementation of some control action that

[2] We avoid calling this system simply "environment", since the latter will refer to the concept of environment in the RL framework.

[3] We refer to ref. [1] for a converse proof: if PBC is not achieved, there exists a passive s-environment destabilizing the controlled system.

17518652, 0, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/cth2.12558 by University Of Twente FEZ, Wiley Online Library on [07/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

fulfils the execution of some task, and at the same time meeting the desired passivity constraint. This is possible thanks to a suitable power-preserving interconnection between the two systems, defined as follows:

$$\begin{pmatrix} \tau_c \\ u_t \end{pmatrix} = \begin{bmatrix} 0 & \frac{w}{x_c} \\ -\frac{w^\top}{x_c} & 0 \end{bmatrix} \begin{pmatrix} \dot{q} \\ y_t \end{pmatrix}. \tag{8}$$

Here $w \in \mathbb{R}^n$ represents the desired task-dependent control action to be passively implemented. The variable w can be seen as an n-dimensional signal, which comes from outside the energy tank algorithm and modulates the relation expressed by Equation (8) (see Figure 1 for a graphical interpretation). This relation, which is the core of the energy tank algorithm, produces two effects: i) the desired action w is correctly implemented: from the side of the robot (1), one obtains $\tau_c = \frac{w}{x_c} y_t = w$, i.e.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + D(q)\dot{q} + \frac{\partial V(q)}{\partial q} = w + J^\top(q)F; \tag{9}$$

and ii) at every time the mechanical power exiting the robot produced by the actuators $\dot{q}^\top \tau_c$ leaves the energy tank since

$$\dot{V}_c = y_t u_t = -w^\top \dot{q} = -\dot{q}^\top w = -\dot{q}^\top \tau_c. \tag{10}$$

Evaluating the variation of the closed-loop storage function $\mathcal{E} = E + V_c$ one achieves the PBC objective (5), which shows that the interconnection achieves indeed a passive closed-loop system with respect to the s-environmental port. The scalar $V_c(x_c)$ indicates the amount of energy that is still at disposal of the control mechanism implementing the action w before losing the described formal passivity. In fact, the interconnection (8), which is modulated by the tank state, becomes singular when x_c, and thus $V_c(x_c)$ is zero, representing the moment in which it becomes impossible to passively perform the desired action w. To meaningfully implement the energy tank algorithm, it is thus necessary to constantly observe the energy in the tank in order to implement a control action $\bar{w}$, rather than w, where

$$\bar{w} = \begin{cases} w, & \text{if } V_c(x_c) \geq \epsilon \geq 0 \\ 0, & \text{if } V_c(x_c) \leq \epsilon, \end{cases} \tag{11}$$

for some small positive energy level ϵ. In this way, formal passivity of the closed-loop system is recovered completely since the system is just detached from the controller at the moment in which no energy budget is left.

Before discussing the peculiarities and limitations of the described algorithm, we introduce the concept of task energy, that we will denote by e^*, firstly defined in ref. [22], which is the minimum energy in the tank necessary to fulfil passively some task. In other words, if $V_c(x_c(t))|_{t=0} > e^*$, the tank never depletes along the whole task horizon. It is important to notice that e^* does not depend on the task only, but also on the specific tank dynamics which is chosen: the choice u_t in Equation (8) is not unique, and many variations have been proposed [5] (e.g. recirculation in the tank of dissipation) which would change the value of task energy for the same task.

2.1.3 | Comments and limitations

i) The energy tank algorithm constitutes a clever way to disembody from any physical dynamics the implementation of a passive control action. In fact, there is no need to design the controller in a way that it mimics a physical system (e.g. in the PD case the controller can be seen as a control spring and damper): as long as a task is achieved through a control input w and e^* is finite, a passive implementation is possible by means of the energy tank algorithm, just setting $V_c(x_c(t))|_{t=0} > e^*$; ii) The aforementioned initial energy assignment is a degree of freedom in the algorithm, whose implications are often naively addressed. In fact, a very high (yet bounded) energy initialization in the tank would technically still fulfil the PBC objective (5), yet creating so-called "practically unstable behaviours" [5, 17]. As a consequence, a naive tank design de facto makes robust stability a property that is not connected to any safety guarantee. In fact, notice that till the energy in the tank is not depleted, the control input w is completely transparent to the tank algorithm, which reduces to a trick to formally prove passivity, with limited significance in the context in which tasks need to be performed in unstructured scenarios. The fact that the mechanical power flow is undefined in sign (i.e. when $\dot{q}^\top \tau$ the tank gets refilled) worsens this criticality, which is sometimes addressed with empirical tank saturation arguments.

To conclude the discussion, both the task energy e^* and the control action w are often difficult to be determined a priori, and these are not independent variables. For complex task executions, it is reasonable to take advantage of simulations to optimize for both variables in a combined way. If the PBC objective can be naively achieved just by initializing the energy in the tank to a sufficiently high value, what we claim to be a useful system theoretic property in the energy tank context is the achievement of the PBC objective combined with an estimation of the task energy. In fact, the knowledge of the latter would lead to a meaningful energy tank initialization, so that a depletion of the tank can be used as a diagnostic tool to detect an important divergence from nominal task execution, beyond formally achieving a passive closed-loop system.

2.2 | Reinforcement learning

Reinforcement learning (RL) [23] is a model-free framework, consisting of an agent interacting with an environment, to solve optimal control problems stated as Markov decision processes (MDPs). MDPs are a mathematical formulation of decision-making in situations where outcomes are partly stochastic and partly under the control of a decision-maker. A MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, p, r)$: at any given state $s_k \in \mathcal{S}$ at time step k, the agent chooses and executes an action $a_k \in \mathcal{A}$ according to a learnt policy $\pi(a_k|s_k)$, then the environment transitions to a new state $s_{k+1} \in \mathcal{S}$ with the unknown state

17518652, 0, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/cth2.12538 by University Of Twente FEZ, Wiley Online Library on [07/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

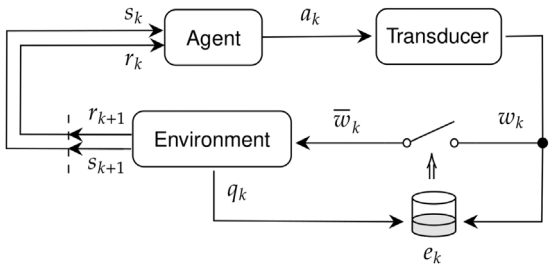


FIGURE 2 RL with energy tank passivization.

transition probability $p(s_{k+1}|s_k, a_k)$ and produces a reward $r_k = r(s_k, a_k)$. An important condition that characterizes the MDPs is that the state transition probability satisfies the Markov property $p(s_{k+1}|s_k, a_k) = p(s_{k+1}|s_k, a_k, \dots, s_1, a_1)$ for any trajectory $s_1, a_1, \dots, s_k, a_k$, which means that the environment is memoryless because the transition to the next state depends only on the current state and action. The Markov property is usually fundamental to guarantee the convergence to the optimal policy in RL algorithms [23]. The goal of an RL algorithm is to find an optimal policy π^* that maximizes the expected γ-discounted sum of future rewards (that we define as return).

While the proposed framework could be combined with any RL algorithm, benchmarking is out of the scope of this work, for this reason in the rest of the paper we only consider the off-policy soft actor-critic (SAC) algorithm [24] with continuous action spaces.

3 | ENERGY TANKS MEET RL

The arbitrariness of the control input w and its disembodiment from any physical dynamics are tempting motivations to choose it as a decision variable in an optimization framework. Indeed, the output of an RL control policy can be directly mapped to the torques w commanded at an actuator level, as shown in Figure 2. This map can be trivially the identity (in case the generalized forces are directly learned in the RL scheme) or be represented by a transducer (e.g. an internal PID controller that maps a position or velocity reference to the commanded torques w), which is often shown to critically improve sample efficiency in the RL framework [25]. No specific transducer between the action space and w is required to implement the proposed scheme, as long as the commanded torques w are accessible. Since MDPs are discrete-time processes, we start by exploiting a simple but powerful result that allows to reproduce the energy tank algorithm at the discrete-time level.

3.1 | The energy sampling approach

Let us denote e_k the level of energy in the tank at the discrete time step k, i.e. $e_k = V_c(kT)$. The energy in the tank at the next time step, initialized by e_0, is updated according to the rule

$$e_{k+1} = e_k - \Delta e_{k+1} \tag{12}$$

where Δe_{k+1} is the amount of energy exiting the tank at time $k+1$, which approximates, at a discrete-time level, the energy leaving the tank in a sampling interval with duration T, i.e. $\int_{Tk}^{T(k+1)} -\dot{V}_c(s)\,\mathrm{d}s$. With a look at the continuous energy balance (10), this energy equals the sum of energies used by each actuator of the robot in the time interval $[Tk, T(k+1))$:

$$\int_{Tk}^{T(k+1)} -\dot{V}_c(s)\mathrm{d}s = \int_{Tk}^{T(k+1)} w(s)^\top \dot{q}(s)\mathrm{d}s.$$

Now, assuming the torque signal is constant with value w_k along the k-th sampling interval, and indicating with $q_k := q(Tk)$, we can further massage the previous expression to define Δe_{k+1} as:

$$w_k^\top \int_{Tk}^{T(k+1)} \dot{q}(s)\mathrm{d}s = w_k^\top (q_{k+1} - q_k) =: \Delta e_{k+1}. \tag{13}$$

It is worth remarking that, in the common conditions of constant torque along the sampling time interval and position sensor collocated to the motor, such defined quantity produces the exact amount of injected energy by the actuators independently of the value of T, and not an estimate. This property makes the connection between the discrete-time RL framework and the energy tank framework particularly appealing since the tank algorithm can exactly be reproduced at a discrete time level without the need of integrating tank dynamics (6) with the input defined in Equation (8).

The definition of Δe_{k+1} in Equation (13) exactly reproduces at a discrete time level the tank algorithm as reviewed in Section 2, but this choice is not unique, and variations are possible on the basis of the passivity margin that one wants to achieve for the closed-loop system. In the sequel of this work, we want to prevent refilling of the tank, and thus use the following update rule

$$\Delta e_{k+1} := \begin{cases} w_k^\top (q_{k+1} - q_k) & \text{if } w_k^\top (q_{k+1} - q_k) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Such an update rule physically represents the impossibility of recovering negative mechanical energy flowing into the system, which is a reasonable assumption for many mechanical systems. Furthermore, this choice serves best our purpose of interpreting the task energy as a metabolic metric to assess an initial energy budget: the energy in the tank can only decrease along the task execution.

In this discrete-time implementation of energy tanks, we aim to achieve the following finite difference version of the PBC objective (5). We indicate $E_k = E(kt)$ the sampled mechanical energy of the robot and $\mathcal{E}_k = E_k + e_k$ the sampled closed-loop storage function.

Sampled PBC objective. *Given the system (1) with the split (3) inducing power balance (4), design a control policy inducing desired control*

17518652, 0, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/cth2.12558 by University Of Twente FEZ, Wiley Online Library on [07/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

action $\tau_c = w_k$ *such that the condition*

$$\mathcal{E}_{k+1} - \mathcal{E}_k \leq \int_{Tk}^{T(k+1)} \dot{x}(s)^\top F(s) ds \tag{15}$$

is verified $\forall k$*, i.e. that the sampled difference of closed-loop storage function is bounded by the energy injected in a sampling interval by the s-environment through the port* $(F, \dot{x})$.

Theorem. *Assuming constant generalized forces* w_k *in the sampling interval* $[Tk, T(k+1)]$*, the PBC objective (15) is achieved using the tank update rule (14).*

Proof. Define $R_{k+1} := \int_{Tk}^{T(k+1)} \dot{q}(s)^\top D(q(s))\dot{q}(s)\mathrm{d}s$ the dissipated power by friction in a sampling interval. Using Equation (12) and integrating Equation (4) in a sampling interval, we compute $\mathcal{E}_{k+1} - \mathcal{E}_k = E_{k+1} - E_k + e_{k+1} - e_k = -R_{k+1} + w_k^\top (q_{k+1} - q_k) + \int_{Tk}^{T(k+1)} \dot{x}(s)^\top F(s)\mathrm{d}s - \Delta e_{k+1}$, where we used the fact that $\tau_c = w_k$ is constant in a sampling interval. The final claim (15) follows from Equation (14) and $R_{k+1} \geq 0$. □

Note that the update rule (14) induces a stronger passivity margin with respect to Equation (13) since when $\Delta e_{k+1} = 0$ the term $w_k^\top (q_{k+1} - q_k)$ is negative, and as such acts as dissipation.

We define the energy spent at step k as the sum of the energies exiting the tank up to that step:

$$\hat{e}_k := \sum_{l=0}^{k-1} \Delta e_{l+1}. \tag{16}$$

Since with the update rule (14) the sequence $\hat{e}_k$ is monotonically non decreasing, the task energy e^* over a task with N time steps can be simply calculated as

$$e^* = \hat{e}_N. \tag{17}$$

We describe now two procedures combining the tank and the RL algorithm. The first one aims at passivizing a control policy that was previously learned agnostically to any passive design. The second exploits the tank architecture also in the training phase and produces by construction a passive learned policy. In Algorithms 1 and 2, we present the pseudocode for the inference and training phases, respectively. Notably, within Algorithm 2, we make use of a generic RL algorithm, which is responsible for handling the policy update and action prediction. This implies that the frequency of policy updates and the action selection, which typically incorporates an exploration strategy, may vary depending on the specific RL algorithm employed.

ALGORITHM 1 Passivizing learned policies

Given: π^* non-passive policy (arbitrary)
$e^* \leftarrow$ estimate task energy using π^*
initialize tank with $e = e_0 \geq e^*$
$q \leftarrow$ get joints coordinates
initialize $q' = q$ and $w = w' = 0$
for *N steps* **do**
  $s \leftarrow$ get state
  $q \leftarrow$ get joints coordinates
  $\Delta e \leftarrow \max\{w'^\top (q - q'), 0\}$
  $e \leftarrow e - \Delta e$
  $w \leftarrow$ predict action with $\pi^*(s)$
  $\bar{w} = w$ if $e \geq \varepsilon$, zero otherwise
  execute $\bar{w}$
  $w' \leftarrow w$
  $q' \leftarrow q$

ALGORITHM 2 Learning passive policies

Given: initial energy in the tank e_0 (arbitrary)
for *M episodes* **do**
  initialize tank with $e = e_0$
  $q \leftarrow$ get joints coordinates
  initialize $q' = q$ and $w = w' = 0$
  for *N steps* **do**
    $s \leftarrow$ get state
    $q \leftarrow$ get joints coordinates
    $\Delta e \leftarrow \max\{w'^\top (q - q'), 0\}$
    $e \leftarrow e - \Delta e$
    $w \leftarrow$ predict action with policy $\pi(s)$
    if $e < \varepsilon$ **then**
      break loop
    else
      execute w
      update π
    $w' \leftarrow w$
    $q' \leftarrow q$

3.2 | Passivization in inference (passivizing learned policies)

We indicate with the term inference the phase where the training is finished and the agent implements the learnt policy, without exploring anymore. The combination of the energy tank architecture and the reinforcement learning technique in inference is quite straightforward, as an RL agent represents in this context an arbitrary controller that the tank algorithm is able to passivize. In fact, we can use any previously trained agent in a generic environment and passivize the controlled system in inference by wrapping the control action with Equation (11), where the energy in the tank is updated according to Equations (12) and (14). Furthermore, an agent able to fulfil a given task in a generic environment with a specific task energy e^*, is also able to passively fulfil that task in an environment where the control action is wrapped with Equation (11) and the energy tank is initialized with a value greater than e^*. This represents indeed the only design requirement to

17518652, 0, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/cth2.12558 by University Of Twente FEZ, Wiley Online Library on [07/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

preserve the agent's performance in executing the task with the learnt policy while achieving formal passivity (15). In order to equip the controlled system with a meaningful passivity property, it is important to initialize the tank with an amount of energy that slightly exceeds e^*, which can be estimated by running the agent without Equation (11) for a number of episodes and measuring the maximum consumption of energy along all the episodes. The proposed procedure provides a constructive way to achieve the PBC objective and brings the additional benefit of task energy estimation, which produces a meaningful closed-loop passivization. This step provides a diagnostic tool indicating a non-regular task execution in case of tank depletion, which is a piece of extra information that a naive tank design lacks.

Note. *It is meaningful to compare the described procedure with the works [17, 18], which aim at passivizing an already given control input, and as such can also be seen as a passivization in inference algorithm. In refs. [17, 18] the task energy is not explicitly estimated and as a consequence the initialization of the tank (whose non-depletion constraint in an optimization problem is what produces the passive approximation) is arbitrary. By solving the optimization with different tank initialization, in ref. [17] different passive approximations are achieved. Here instead, motivated by the fact that passivity can be achieved anyhow for any finite task energy and that the learnt policy represents the way the task is optimally executed, we first estimate the task energy to constraint the degree of freedom of tank initialization.*

3.3 | Passivization in training (learning passive policies)

In many circumstances, it might be desirable to continuously achieve a passive closed-loop system also in the training phase, i.e. when the task-dependent control policy is being learned. This is desirable for instance when the training takes place in real life, and not in simulations, such that initial exploration phases are guaranteed not to undergo hazardous unbounded energy generation. Furthermore, the passivized training phase endows the agent with awareness of the metabolic spending encoded in the tank architecture, induced by the initial energy budget e_0. As a consequence, the learned policy will be influenced by the tank initialization e_0, and in particular, the task energy e^* will be directly learned together with the control policy in a combined way, strengthening the significance of the resulting passivity property. This mechanism provides a clear biomimetic perspective to the scheme since the energy budget and the task-dependent policy are not independent variables.

The proposed procedure is achieved by implementing the learning in the extended RL framework depicted in Figure 2., i.e. the standard RL scenario with the tank algorithm embedded. We recognize the following technical issue which needs to be addressed before implementing the training.

3.3.1 | Loss of Markovian property

Notice that the current value of the tank state e_k depends on the entire state-action trajectory of the system from the beginning to step k. This is due to the update rule (12), where Δe_{k+1} depends both on the joint position q_k (which in most robotics applications is part of the RL state s_k) and the action a_k.

Furthermore the value e_k is able, due to Equation (11), to influence the future dynamics of the environment in case of tank depletion. For these reasons the new environment that the agent perceives by including also the energy tank architecture as in Figure 2 does not preserve the memorylessness of the original environment, or in other words it loses the Markov property.

If the Markov property is not satisfied, most RL algorithms lose formal proof of convergence to the optimal policy, and long-term prediction performance can degrade when the one-step predictions defining the Markov property become inaccurate [26]. We identify two solutions to restore the Markov property.

3.3.2 | Extended state

A common practice when the environment is non-Markovian because a relevant time-dependent part of the information is missing from the agent's state but accessible is simply to include it in the state [26]. However, under this formulation, the agent might require a dedicated tuning process to maintain similar performance.

3.3.3 | Extended termination

As an alternative, the episode's termination condition can be extended to the situation in which the energy in the tank depletes. In case a robotic platform is involved, terminating the episode would mean breaking the joints, instead of giving zero torque as done in the original formulation. After the termination, the environment is reset and a new episode can start with a restored level of energy in the tank. In this formulation, the Markov property is restored without changing the state and the reward. In fact, the termination of the episode removes the switch behaviour (11), and the agent never meets a state influenced by the free dynamics.

The reward function has to be non-negative, since the length of an episode is not fixed. In fact, using a negative reward, the agent might attempt to maximize the cumulative return by learning actions leading to tank depletion.

4 | SIMULATIONS

We consider two environments implemented in MuJoCo physics simulator [27]:

17518652, 0, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/cth2.12538 by University Of Twente FEZ, Wiley Online Library on [07/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

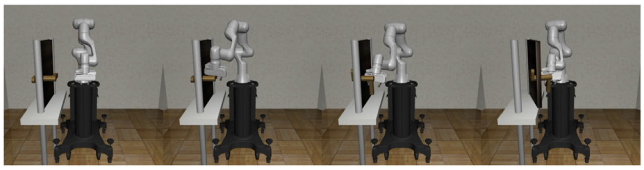


FIGURE 3 `DoorOpening` environment.

1) `DoorOpening`: Here, we adopted an implementation from Robotsuite simulation framework [11] where a 7-DoF robotic arm must learn to open a door by turning the handle as shown in Figure 3. The door location is randomized at the beginning of each episode. The robot is a Panda Franka Emika which mounts a parallel-jaw gripper equipped with two small finger pads. The torques provided to the 7 joints of the robot are generated using a proportional control law $\tau = k_p(\dot{q}_d - \dot{q})$, with $k_p = 0.03$, that follow a target joint velocity $\dot{q}_d$ provided by the agent at 50 Hz.
2) `Pendulum`: In this environment we created a pendulum composed of a rod suspended by one extremity from a pivot actuated by a motor controlled in torque with a control frequency of 50 Hz. The rod is subject to gravity while friction loss is present in the joint. Let us denote the joint angle as β. In the reference configuration $\beta = 0$ the pendulum heads horizontally right. At the beginning of each episode, the angle is randomly initialized $\beta \sim \mathcal{N}(-\pi/2, 0.05\pi)$. The environment state is defined as the vector $s_k = [\sin\beta_k \quad \cos\beta_k \quad \tanh\dot{\beta}_k]^\top$, while the agent action corresponds to the torque applied to the joint. Since the inverted pendulum problem consists of bringing the rod to the inverted position ($\sin\beta^* = 1$) and holding it there as long as possible, we design a positive reward function $r_k = (1 + |\sin\beta_k - 1| + 0.1|\tanh\dot{\beta}_k| + 0.01|\tau_k|)^{-1}$ which is inversely proportional to a weighted sum of the absolute values of position error ($\sin\beta_k - 1$), velocity error ($\tanh\dot{\beta}_k$) and torque (τ_k). These last two components help to remove oscillatory behaviours in the transition phase and at the equilibrium point.

In all the simulations presented in this work the agents are trained using SAC algorithm implemented in PyTorch and trained with an NVIDIA GeForce GTX 1080 Ti on an Intel Core i7-7700K CPU clocked at 4.20GHz. In Table 1 the training configurations for both environments are reported. Additionally, generalized State-dependent exploration (gSDE) is employed, where a new noise matrix is sampled every episode. The entropy regularization coefficient is learned automatically as done in ref. [28].

In the rest of this section, we expose the results obtained from the simulations run on the two environments introduced above. In particular, the passivization of a policy learned on the `DoorOpening` and the learning of a passive policy on `Pendulum` are discussed respectively in Sections 4.1 and 4.2. In Table 2 the notation adopted for each agent in training and inference is briefly reported.

TABLE 1 Hyperparameters.

	`DoorOpening`	`Pendulum`
Actor learning rate	0.01	0.005
Critic learning rate	0.0005	0.005
Soft update coefficient	0.005	0.003
Batch size	128	256
Number of epochs	2000	200
Steps per epoch	2500	2500
Steps per trajectory	500	500
Steps before training	3300	2500
Target update period	5	5
Gradient steps	1000	500
Neurons per layer	[256,256]	[256,256]

TABLE 2 Notation.

ϕ_∞	$\triangleq$	Agent trained without passivization
ϕ_{e_0}	$\triangleq$	Agent trained with passivization and tank initialized at e_0
$\phi_{\infty\|\infty}$	$\triangleq$	Agent ϕ_∞ in inference without passivization
$\phi_{\infty\|e^*}$	$\triangleq$	Agent ϕ_∞ in inference with passivization and tank initialized at e^*
$\phi_{e_0\|e^*}$	$\triangleq$	Agent ϕ_{e_0} in inference with passivization and tank initialized at e^*
$\phi_{\infty\|\infty,\delta}$	$\triangleq$	Agent ϕ_∞ in inference with an external force field and without passivization
$\phi_{\infty\|e^*,\delta}$	$\triangleq$	Agent ϕ_∞ in inference with an external force field, passivization, and tank initialized at e^*

4.1 | Passivization in inference

To study the effects of passivization in inference, an external field of force is applied to the system as a way to perturb the nominal task. The magnitude of the force is chosen in order to let the tank deplete when it is initialized with an amount of energy corresponding to the task energy measured in the nominal case. For this experiment, we consider the `DoorOpening` environment and we employ a pre-trained model available online[4]. As visible in Figure 4, under the effect of the external force the agent without the passivization in inference $\phi_{\infty|\infty,\delta}$ can spend an unbounded amount of energy, while the consumption is limited when the same agent is wrapped with passivization $\phi_{\infty|e^*,\delta}$. The energy tank is initialized with the task energy e^* measured as the maximum $\hat{e}_k$ over the 100 episodes in the agent without the external force and without passivization $\phi_{\infty|\infty}$. From this simulation, we can better comprehend and appreciate the versatility of the proposed framework. In fact, an agent trained by a third party to optimize the process in a non-passive way is readily passivized by wrapping the

[4] https://github.com/ARISE-Initiative/robosuite-benchmark

17518652, 0, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/cth2.12538 by University Of Twente FEZ, Wiley Online Library on [07/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

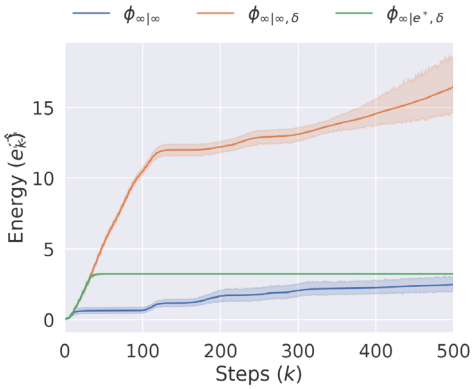


FIGURE 4 Average energy spent in `DoorOpening` in inference.

decision variable as detained in Section 3.2 and without the need to retrain/modify the agent.

4.2 | Passivization in training

Let us consider two agents ϕ_∞ and ϕ_{e_0} trained in the `Pendulum` environment sharing the same hyperparameter configuration, but ϕ_{e_0} is passivized with the framework of Section 3.3 and the *Extended Termination* method. For each simulation on the `Pendulum`, 5 instances with different random seeds for the pseudo-random generators are trained. Implementing the training with a sufficiently low e_0 provides some level of meaningful robust stability in the training phase. Furthermore, we aim to analyze how the energy budget e_0 imposed in training influences the learned policy and the task energy e^*, which are correlated variables.

We choose as e_0 the task energy estimated from ϕ_∞, measured in inference as the maximum $\hat{e}_k$ over 100 episodes. As visible in Figure 5, the task energy e^* estimated by ϕ_{e_0} is lower than the energy that the system spends to perform the task when using the policy ϕ_∞. Indeed we appreciate how the agent ϕ_{e_0} learns to perform the task with lower energy consumption. Furthermore, the average return of the two agents during the training visible in Figure 5 converges to the same value (almost 2500), while ϕ_∞ can arrive to spend a level of energy that is almost 50 times greater than the one spent in ϕ_{e_0}, see Figure 5.

Clearly, the episode terminations caused by the energy constraints (i.e. when the tank depletes), disturb the exploration during the training, which results in a slower convergence to the final policy. Figure 5 shows a comparison, in inference, of the two agents where a measure of the position error $d_k = 1 - \sin(\beta_k)$ is adopted as a success metric of the task. As visible both the agents converge to the same error, with a smoother trajectory for ϕ_{e_0}.

5 | CONCLUSIONS AND FUTURE WORK

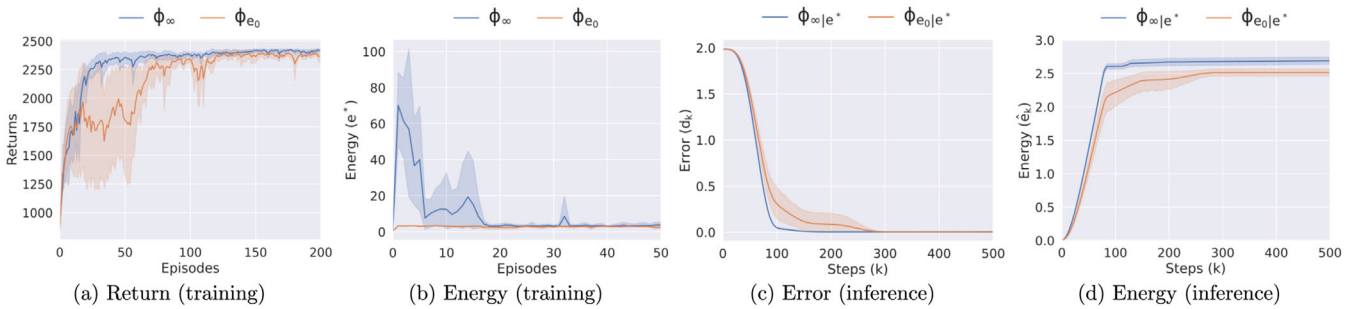
We introduced a framework merging the energy tank algorithm, used as a tool to passivize arbitrary control schemes, and reinforcement learning, representing the most versatile method to learn control policies along complex tasks. The presented procedures allow to passivize any control policy, and to constructively learn passive policies, tackling the problem represented by a lack of optimization in the design phase of passive controllers. As future work, thinking beyond merely passive designs, we are processing a throughout study of energy-aware robotics in the proposed neural framework, exploiting the energy tank architecture to embed metabolic and safety metrics. Furthermore, we are working towards an experimental validation of the proposed schemes, with the goal of passivizing RL policies applied to robotic system both in training and inference.

AUTHOR CONTRIBUTIONS

Riccardo Zanella: Conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing - original draft, writing - review and editing. Gianluca Palli: Funding acquisition, project administration, resources. Stefano Stramigioli: Supervision. Federico Califano: Conceptualization, formal analysis, investigation, methodology, supervision, writing - original draft, writing - review and editing.

ACKNOWLEDGEMENTS

This work was supported by the European Commission's Horizon Europe Programme with the project IntelliMan - AI-Powered Manipulation System for Advanced Robotic Service, Manufacturing and Prosthetics - under grant agreement No 101070136.



(a) Return (training) (b) Energy (training) (c) Error (inference) (d) Energy (inference)

FIGURE 5 In (a) and (b) respectively the average returns obtained and the energy spent in training. In (c) and (d) respectively the average position error and energy spent in inference.

17518652, 0, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/cth2.12538 by University Of Twente FEZ, Wiley Online Library on [07/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Data available on request from the authors.

ORCID

Riccardo Zanella https://orcid.org/0000-0001-5764-9896

REFERENCES

1. Stramigioli, S.: Energy-Aware Robotics. In: Lecture Notes in Control and Information Sciences. Vol. 461, pp. 37–50. Springer, Cham (2015)
2. van der Schaft, A.: L2-Gain and Passivity Techniques in Nonlinear Control (3rd ed.). Springer, Cham (2016)
3. Duindam, V., Macchelli, A., Stramigioli, S., Bruyninckx, H.: Modeling and control of complex physical systems. Springer, Berlin, Heidelberg (2009)
4. Ortega, R., García-Canseco, E.: Interconnection and damping assignment passivity-based control: a survey. Eur. J. Control 10(5), 432–450 (2004)
5. Califano, F., Rashad, R., Secchi, C., Stramigioli, S.: On the use of energy tanks for robotic systems. In: Human-Friendly Robotics 2022, pp. 174–188. Springer, Cham (2023).
6. Secchi, C., Stramigioli, S., Fantuzzi, C.: Position drift compensation in port-Hamiltonian based telemanipulation. In: IEEE International Conference on Intelligent Robots and Systems, pp. 4211–4216. IEEE, Piscataway, NJ (2006)
7. Duindam, V., Stramigioli, S.: Port-based asymptotic curve tracking for mechanical systems. Eur. J. Control 10(5), 411–420 (2004)
8. Dimeas, F., Aspragathos, N.: Online stability in human-robot cooperation with admittance control. IEEE Trans. Haptic 9(2), 267–278 (2016)
9. Zeng, A., Song, S., Welker, S., Lee, J., Rodriguez, A., Funkhouser, T.: Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4238–4245. IEEE, Piscataway, NJ (2018)
10. Singh, A., Yang, L., Finn, C., Levine, S.: End-to-End Robotic Reinforcement Learning Without Reward Engineering. University of California, Berkeley, CA (2019)
11. Zhu, Y., Wong, J., Mandlekar, A., Martín-Martín, R.: robosuite: a modular simulation framework and benchmark for robot learning. arXiv:2009.12293 (2020)
12. Buşoniu, L., de Bruin, T., Tolić, D., Kober, J., Palunko, I.: Reinforcement learning for control: Performance, stability, and deep approximators. Annu. Rev. Control 46, 8–28 (2018)
13. Shahriari, E., Johannsmeier, L., Haddadin, S.: Valve-based virtual energy tanks: a framework to simultaneously passify controls and embed control objectives. In: Proceeding of the American Control Conference 2018, pp. 3634–3641. IEEE, Piscataway, NJ (2018)
14. Benzi, F., Brunner, M., Tognon, M., Secchi, C., Siegwart, R.: Adaptive Tank-based control for aerial physical interaction with uncertain dynamic environments using energy-task estimation. IEEE Rob. Autom. Lett. 7(4), 9129–9136 (2022)
15. Shahriari, E., Johannsmeier, L., Jensen, E., Haddadin, S.: Power flow regulation, adaptation, and learning for intrinsically robust virtual energy tanks. IEEE Rob. Autom. Lett. 5(1), 211–218 (2020)
16. Califano, F., van Dijk, D., Roozing, W.: A task-based post-impact safety protocol based on energy tanks. IEEE Rob. Autom. Lett. 7(4), 8791–8798 (2022)
17. Benzi, F., Ferraguti, F., Riggio, G., Secchi, C.: An energy-based control architecture for shared autonomy. IEEE Trans. Rob. 38(6), 3917–3935 (2022)
18. Capelli, B., Secchi, C., Sabattini, L.: Passivity and control barrier functions: optimizing the use of energy. IEEE Rob. Autom. Lett. 7(2), 1356–1363 (2022)
19. Massaroli, S., Poli, M., Califano, F., Park, J., Yamashita, A., Asama, H.: Optimal energy shaping via neural approximators. SIAM J. Appl. Dyn. Syst. 21(3), 2126–2147 (2022)
20. Sprangers, O., Babuška, R., Nageshrao, S.P., Lopes, G.A.D.: Reinforcement learning for port-hamiltonian systems. IEEE Trans. Cybern. 45(5), 1017–1027 (2015)
21. Nageshrao, S.P., Lopes, G.A.D., Jeltsema, D., Babuška, R.: Passivity-based reinforcement learning control of a 2-DOF manipulator arm. Mechatronics 24(8), 1001–1007 (2014)
22. Schindlbeck, C., Haddadin, S.: Unified passivity-based Cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 440–447. IEEE, Piscataway, NJ (2015)
23. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (2018)
24. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: [SAC] Soft actor-critic. In: 35th International Conference on Machine Learning, ICML 2018, pp. 2976–2989. Microtome Publishing, Brookline, MA (2018)
25. Smith, L., Kostrikov, I., Levine, S.: A walk in the park: learning to walk in 20 minutes with model-free reinforcement learning. arXiv:2208.07860 (2022)
26. Whitehead, S.D., Lin, L.J.: Reinforcement learning of non-Markov decision processes. Artif. Intell. 73(1–2), 271–306 (1995)
27. Todorov, E., Erez, T., Tassa, Y.: MuJoCo: A physics engine for model-based control. In: IEEE International Conference on Intelligent Robots and Systems, pp. 5026–5033. IEEE, Piscataway, NJ (2012)
28. Haarnoja, T., et al.: Soft actor-critic algorithms and applications. arXiv:1812.05905 (2018)

How to cite this article: Zanella, R., Palli, G., Stramigioli, S., Califano, F.: Learning passive policies with virtual energy tanks in robotics. IET Control Theory Appl. 1–10 (2024). https://doi.org/10.1049/cth2.12558

17518652, 0, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/cth2.12558 by University Of Twente FEZ, Wiley Online Library on [07/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License