

Passive Interaction Control With Dynamical Systems

Klas Kronander and Aude Billard

Abstract—Autonomous dynamical systems (DS) has emerged as an extremely flexible and powerful method for modeling robotic tasks. Task execution of DS models is typically done in an open-loop manner in combination with standard low level controller, e.g. position controller or impedance controller. Such an arrangement has two important drawbacks: 1) it is not **passive** and 2) the DS model cannot respond to physical perturbations on the robot body. These can imply severe consequences in motion tasks involving expected or unexpected contact with objects whose position is **unknown and dynamic**. We propose a novel control architecture that **closes the loop** around the DS, ensures passivity and allows intuitive tuning of the robot impedance. We evaluate our approach in a comparative study in an **uncertain manipulation task with unexpected contact**.

Index Terms—Compliance and Impedance Control, Motion Control of Manipulators, Physical Human-Robot Interaction.

I. INTRODUCTION

DYNAMICAL Systems (DS) has emerged as a general and highly flexible means of representing robot motions. It has been demonstrated that many of the proposed DS formulations lend themselves well to learning, both in a supervised setting as well as reinforcement learning [1]–[3]. Qualitative properties such convergence to a limit cycle or stability at an attractor point can be ensured regardless of the data provided to the learning algorithm [4]–[7]. Furthermore, it is possible to incorporate features such as joint or task level object avoidance into DS formulations [8], [9].

In parallel to the development of DS-based learning systems, the field of robotic manipulation has in recent years seen a revitalized interest in control of mechanical interaction, a topic which largely rests upon foundations of Hogans impedance control formulation [10]. To use a DS task representation with an impedance controller, it is necessary to integrate the DS over time to yield a reference trajectory, see Fig. 1 left. In such a configuration, the reactivity of the DS is used with respect to perturbations that are captured using external sensors, e.g. the pose estimate of a moving target point or obstacles. However, the full power of the DS is not used, since the integration of the reference trajectory disallows the DS to react to physical perturbations on the robot. For physical interaction tasks such reactivity can be crucial, and it would hence be desirable to instead use a controller that feeds back the actual state of the robot to the DS, see Fig. 1, right.

Manuscript received August 31, 2015; accepted December 8, 2015. Date of publication December 17, 2015; date of current version January 11, 2016. This paper was recommended for publication by Associate Editor A. M. Zanchettin and Editor P. Rocco upon evaluation of the reviewers' comments. This work was supported by the European Union Seventh Framework Programme FP7/2007–2013 under Grant 288533 ROBOHOW.COG.

The authors are with the Learning Algorithms and Systems Laboratory (LASA), Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland (e-mail: klas.kronander@epfl.ch; aude.billard@epfl.ch).

Digital Object Identifier 10.1109/LRA.2015.2509025

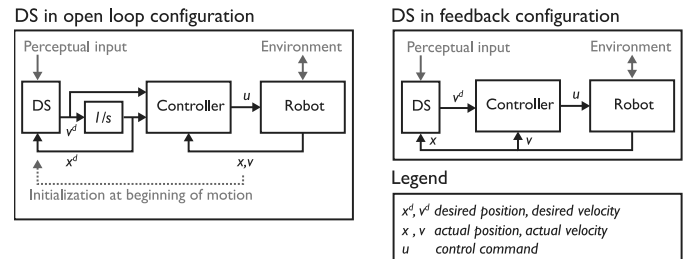


Fig. 1. Illustration of open loop and closed loop control configurations with DS. In the open loop configuration, the DS is updated with the desired position resulting from integration of the desired velocity. The actual position of the robot is only used for initializing the integration at the beginning of the motion. In contrast, the feedback configuration continuously updates the DS with the actual position and realizes a control on the velocity error.

An important property for controllers interacting with unknown environments is **passivity**. A controller that ensures a passive relation between external forces and robot velocity will yield stable behavior in free motion and in contact with any passive environment [11]. In this sense, classic impedance control is only passive in the regulation case and the passivity can no longer be ensured if the desired velocity is non-zero. The loss of passivity during tracking is an important drawback of impedance control and a problem that arises in any controller driven by time-indexed reference trajectories. A thorough analysis of this problem is provided in [12], which advocates to tackle it by encoding tasks using time-independent velocity fields (first order DS) and proposes a controller that allows tracking of desired DS. Related work has proposed a similar approach for passivity in the curve tracking problem [13]. These works exploit a time-independent encoding of the task to ensure passivity and energy-efficient, accurate tracking of the DS. The closed-loop dynamics are however rather complicated and the specification of a mechanical impedance is not clear or easy to specify. Another related solution was presented in [14], which works with the classical trajectory-based approach but constructs a tangent vector field and scales the velocity as necessary to ensure passivity.

In this work, **we aim to combine the advantages of impedance control and a passive control system without dependency on time**. In contrast to [12] and [13] which are based on redistribution of kinetic energy along the desired direction of motion, we propose a control structure which is based on selective dissipation of energy in directions that are irrelevant to the task. As we shall see, this allows to easily tune the mechanical impedance while ensuring passivity. In recent years, many researchers have introduced various learning systems for time-independent DS motion models [2], [6], [5], [7]. The focus in these works has been on the ability to learn from example trajectories and to ensure that integration of resulting models

satisfied the topological desiderata, e.g. boundedness or stability. However, how these systems should be used in closed-loop control has not been addressed. In this paper, we introduce a new controller that allows to use such learned DS task models in a closed-loop configuration and at the same time ensures stable interaction with any passive environment. Summarizing, the main contributions of this paper are:

- 1) A design methodology for a damping matrix whose directionality varies as a function of the desired velocity.
- 2) Using the damping design to allow passive velocity feedback control in conservative DS.
- 3) Applying **the concept of energy tanks** to allow velocity feedback control when the desired dynamics are non-conservative systems.
- 4) An analysis of the link to classical impedance control and impedance specification of the proposed controller.

We evaluate our controller in a **robotic reaching task with limited knowledge on 1) the robot dynamics and 2) the environment**. We show that the proposed controller has advantages over classical impedance control both in respecting the shape of the desired motion as well as keeping forces low in unexpected contact.

II. PROBLEM STATEMENT

Let $f(\xi)$ be a Dynamical System describing a nominal motion plan for a robotic task. The variable ξ represents a generalized state variable, which could be e.g. robot joint angles or Cartesian position. Any integral curve of f represents the desired motion of the robot in the absence of perturbations. In this work, we make the following assumptions on the task dynamics:

- 1) It is known, we have access to it and can evaluate $f(\xi)$ for any ξ at any time.
- 2) $f(\xi)$ is a continuous function.
- 3) It has a single equilibrium point ξ^* such that $f(\xi^*) = 0$. Furthermore, ξ^* is a *stable* equilibrium point.

There exist several techniques that allow to learn such stable DS models from demonstrations [6], [5], [15], [7] and the learning of such models is hence not treated in this paper. We consider rigid-body dynamics described in the generalized state variable ξ :

$$M(\xi)\ddot{\xi} + C(\xi, \dot{\xi})\dot{\xi} + g(\xi) = \tau_c + \tau_e \quad (1)$$

The goal of this work is to design a controller τ_c so that Eq. (1) has the following properties:

- 1) **Passivity** (τ_e, ξ) should be preserved for the controlled system.
- 2) The controller should make the robot move according to f , and **dissipate kinetic energy in directions perpendicular to f** .
- 3) It should be possible to **vary task-based impedance** of the manipulator, e.g. how dynamics defining how external forces τ_e affect the velocity $\dot{\xi}$.

In this work, we consider interaction control scenarios. Hence, the goal is to have a passive system, but not necessarily

asymptotically stable¹. Therefore, we do not request that the resulting closed-loop system be asymptotically stable at the attractor, even if the task DS $f(\xi)$ has this property when considered in isolation.

III. PROPOSED APPROACH

A. Selective Energy Dissipation With Task Varying Damping

Consider a feedback controller consisting solely of a damping term and a gravity cancellation term:

$$\tau_c = g(\xi) - D\dot{\xi} \quad (2)$$

where $D \in \mathbb{R}^{N \times N}$ is some positive semi-definite matrix. Obviously, since the nominal model $f(\xi)$ does not appear in Eq. (2), we do not expect that the system would converge to the desired velocity under this simple control. However, this structure does allow to selectively tune the dissipation of energy in desired and undesired directions.

It is easy to show that the controller in (2) renders the system (1) passive with respect to the input τ_e , output $\dot{\xi}$ with the kinetic energy as storage function. This is true for an arbitrarily varying damping, as long as it remains positive semi-definite. We will exploit this fact and construct a varying damping term that dissipates selectively in directions orthogonal to the desired direction of motion given by $f(\xi)$. Let e_1, \dots, e_N be an orthonormal basis for \mathbb{R}^N with e_1 pointing in the desired direction of motion. Hence, let $e_1 = \frac{f(\xi)}{\|f(\xi)\|}$, and let e_2, \dots, e_N be an arbitrary set of mutually orthogonal and normalized vectors. Let the matrix $Q(\xi) \in \mathbb{R}^{N \times N}$ be a matrix whose columns are e_1, \dots, e_N . This matrix is a function of the state ξ , since the vectors e_1 and hence all e_1, \dots, e_N depend on ξ via $f(\xi)$. We then define the **state-varying damping matrix $D(\xi)$** as follows:

$$D(\xi) = Q(\xi)\Lambda Q(\xi)^T \quad (3)$$

where Λ is a diagonal matrix with non-negative values on the diagonal $\lambda_1, \dots, \lambda_N \geq 0$.

By adjusting these damping values, different dissipation behaviors can be achieved. For example, setting $\lambda_1 = 0$ and $\lambda_2, \dots, \lambda_N > 0$ results in a system that selectively dissipates energy in directions perpendicular to the desired motion. Hence, external work in irrelevant directions is opposed while along the integral curves of $f(\xi)$ the system is free to move. If $\|f(\xi)\|$ is very small (e.g. near the equilibrium point), finding the basis for the damping becomes ill-defined. This can easily be handled by keeping the previous basis if $\|f(\xi)\| < \eta$, where $\eta > 0$ is some predetermined small threshold value.

B. Tracking in Conservative DS

While the selective damping in Section III-A allowed selective energy dissipation, it can not drive the robot forward along the integral curves of f . The system would hence only move if external energy is provided to it, in which case kinetic energy along the desired direction of motion would be accepted while

¹Asymptotic stability is neither desirable nor possible in the presence of persistent perturbations.

kinetic energy in directions perpendicular to the desired motion would be dissipated. In order to make the robot move along the integral curves of \mathbf{f} without external input, we have to add some driving control to Eq. (2). This can be achieved through rather simple means, provided that the nominal task model \mathbf{f} is the negative gradient of an associate potential function. This restricted class of DS will be referred to as conservative vector fields in the remainder of this paper.

Definition 1 (Conservative DS): An autonomous dynamical system $\mathbf{f}(\boldsymbol{\xi})$ is conservative if and only if there exists a scalar potential function V_f such that:

$$\mathbf{f}(\boldsymbol{\xi}) = -\nabla V_f(\boldsymbol{\xi}) \quad (4)$$

Consider now a modified controller with negative velocity error feedback:

$$\tau_c = \mathbf{g}(\boldsymbol{\xi}) - \mathbf{D}(\boldsymbol{\xi})(\dot{\boldsymbol{\xi}} - \mathbf{f}(\boldsymbol{\xi})) = \mathbf{g}(\boldsymbol{\xi}) - \mathbf{D}(\boldsymbol{\xi})\dot{\boldsymbol{\xi}} + \lambda_1 \mathbf{f}(\boldsymbol{\xi}) \quad (5)$$

The last equality is due to the fact that $\mathbf{f}(\boldsymbol{\xi})$ is an eigenvector of $\mathbf{D}(\boldsymbol{\xi})$ as described in Section III-A.

Proposition 1: Let $\mathbf{f}(\boldsymbol{\xi})$ be a conservative system with an associated potential function $V_f(\boldsymbol{\xi})$. Then, the system (1) under control given by (5) is passive with respect to the input output pair $\tau_e, \dot{\boldsymbol{\xi}}$ with the storage function $W(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \frac{1}{2}\dot{\boldsymbol{\xi}}^T \mathbf{M}(\boldsymbol{\xi})\dot{\boldsymbol{\xi}} + \lambda_1 V_f(\boldsymbol{\xi})$

Proof: The rate of change of W is:

$$\dot{W}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \dot{\boldsymbol{\xi}}^T \mathbf{M}(\boldsymbol{\xi})\ddot{\boldsymbol{\xi}} + \frac{1}{2}\dot{\boldsymbol{\xi}}^T \dot{\mathbf{M}}(\boldsymbol{\xi})\dot{\boldsymbol{\xi}} + \lambda_1 \nabla V_f^T \dot{\boldsymbol{\xi}} \quad (6)$$

Substituting $\mathbf{M}\ddot{\boldsymbol{\xi}}$ from Eq. (1) with τ_c given by Eq. (5) yields:

$$\begin{aligned} \dot{W}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) &= \frac{1}{2}\dot{\boldsymbol{\xi}}^T (\dot{\mathbf{M}} - 2\mathbf{C})\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}^T \mathbf{D}\dot{\boldsymbol{\xi}} + \dot{\boldsymbol{\xi}}^T \tau_e + \\ &\quad + \lambda_1 \dot{\boldsymbol{\xi}}^T \mathbf{f}(\boldsymbol{\xi}) + \lambda_1 \nabla V_f^T \dot{\boldsymbol{\xi}} \end{aligned} \quad (7)$$

where dependencies of \mathbf{M}, \mathbf{D} on $\boldsymbol{\xi}$ and \mathbf{C} of $\dot{\boldsymbol{\xi}}$ have been omitted for cleanliness of notation. In Eq. (7) the first term is null due to the skew-symmetry of the matrix $\dot{\mathbf{M}} - 2\mathbf{C}$ and the last two terms cancel because $\mathbf{f}(\boldsymbol{\xi}) = -\nabla V_f$. Hence, we have:

$$\dot{W}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = -\dot{\boldsymbol{\xi}}^T \mathbf{D}\dot{\boldsymbol{\xi}} + \dot{\boldsymbol{\xi}}^T \tau_e \quad (8)$$

which shows that the system is passive with $\tau_e, \dot{\boldsymbol{\xi}}$ as input/output pair. ■

It is important to remark that it is never necessary to evaluate the potential function, since it is merely its existence that is required. The controller derived in this section is strongly related to a class of controllers based on energy-shaping, pioneered by [16]. The controller given by Eq. (5) satisfies the requirements from Section II. Unfortunately, only very simple tasks can be modeled with conservative (also called irrotational) DS models, and learned models such as Stable Estimator of Dynamical Systems (SEDS) [5] or Locally Modulated Dynamical Systems (LMDS) [6] are not in general conservative. The accuracy to which the desired dynamics will be followed in absence of external wrench depends on the dynamic properties of the manipulator as well as the desired dynamics.

C. Extension to Nonconservative DS

The restriction to conservative systems previously was necessary to cancel the term $\lambda_1 \dot{\boldsymbol{\xi}}^T \mathbf{f}(\boldsymbol{\xi})$ in the rate of change of the kinetic energy in the system. However, that term only has to be canceled for $\dot{\boldsymbol{\xi}}^T \mathbf{f}(\boldsymbol{\xi}) > 0$. If $\dot{\boldsymbol{\xi}}^T \mathbf{f}(\boldsymbol{\xi}) \leq 0$, the control is contributing to the decrease of kinetic energy. Consider at time t_0 that the system has a particular energy level $W(t_0)$. Now assume that at $t_1 > t_0$ energy has been dissipated so that $W(t_1) < W(t_0)$. This dissipated energy provides in a sense a passivity margin — the system would be passive even if $W(t_1) = W(t_0)$. This fact can be exploited by augmenting the state vector with a virtual state that is capable of storing energy that would otherwise be lost in dissipation. This stored energy can then be released in order to implement control actions that would be non-passive in the original system without the storage element. This concept, sometimes referred to as energy tanks, features prominently bilateral tele-manipulation [17]–[19] and has also been applied to variable stiffness control [20].

Let $\mathbf{f}(\boldsymbol{\xi})$ be decomposed into a conservative part and a non-conservative part:

$$\mathbf{f}(\boldsymbol{\xi}) = \mathbf{f}_C(\boldsymbol{\xi}) + \mathbf{f}_R(\boldsymbol{\xi}) \quad (9)$$

where \mathbf{f}_C denotes the conservative part which has an associated potential function, and \mathbf{f}_R denotes the non-conservative part. Any system can be written in this form, although it may not always be trivial to find such a decomposition. Section III-D gives guidelines as to how such a decomposition can be acquired.

We shall consider an additional state variable $s \in \mathbb{R}$ that represents stored energy. It is a virtual state to which we can assign arbitrary dynamics. We shall consider dynamics coupled with the robot state variables $\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}$ as follows:

$$\dot{s} = \alpha(s)\dot{\boldsymbol{\xi}}^T \mathbf{D}\dot{\boldsymbol{\xi}} - \beta_s(z, s)\lambda_1 z \quad (10)$$

where $z = \dot{\boldsymbol{\xi}}^T \mathbf{f}_R(\boldsymbol{\xi})$. The scalar functions $\alpha : \mathbb{R} \mapsto \mathbb{R}$ and $\beta : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ control the flow of energy between the virtual storage s and the robot, and will be defined in the following. It is necessary to put an upper bound on the virtual storage, such that it can only store a finite amount of energy. Let $\bar{s} > 0$ denote this upper bound. Then, $\alpha(s)$ should satisfy:

$$\begin{cases} 0 \leq \alpha(s) \leq 1 & s < \bar{s} \\ \alpha(s) = 0 & s \geq \bar{s} \end{cases} \quad (11)$$

Disregarding for the moment the second term in Eq. (10), it is clear that the first term (energy that would otherwise be dissipated) only adds to the virtual storage as long as the latter remains below its upper bound, $s < \bar{s}$. Now turning to the second term of Eq. (10), $\beta_s(z, s)$ should satisfy:

$$\begin{cases} \beta_s(z, s) = 0 & s \leq 0 \text{ and } z \geq 0 \\ \beta_s(z, s) = 0 & s \geq \bar{s} \text{ and } z \leq 0 \\ 0 \leq \beta_s(z, s) \leq 1 & \text{elsewhere} \end{cases} \quad (12)$$

Considering the second term in Eq. (10), it is clear that with β_s satisfying Eq. (12), transfer to the virtual storage ($z < 0$) is only

possible as long as $s < \bar{s}$. Conversely, extraction of energy from the storage ($z > 0$) is only possible as long as $s > 0$. When the storage is depleted, the controller can no longer be allowed to drive the system along \mathbf{f}_R if this results in increasing the kinetic energy of the system. Therefore, we introduce the scalar function $\beta_R(z, s)$ whose role is to modify the control signal if the storage is depleted.

$$\tau_c = \mathbf{g}(\xi) - \mathbf{D}\dot{\xi} + \lambda_1 \mathbf{f}_c(\xi) + \beta_R(z, s) \lambda_1 \mathbf{f}_R(\xi) \quad (13)$$

where $\beta_R : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ is a scalar function that should satisfy:

$$\begin{cases} \beta_R(z, s) = \beta_s(z, s) & z \geq 0 \\ \beta_R(z, s) \geq \beta_s(z, s) & z < 0 \end{cases} \quad (14)$$

We are now ready to state the main result of this section.

Theorem 1: Let the nominal task model $\mathbf{f}(\xi)$ be composed of conservative and non-conservative parts according to Eq. (9). Let the system (1) be controlled by Eq. (13) and assume the functions α, β_s, β_R satisfy the conditions in Equations (11), (12) and (14) respectively. Let $0 < s(0) \leq \bar{s}$. The resulting closed loop system is passive with respect to the input-output pair $\tau_e, \dot{\xi}$.

Proof: First, note that $0 < s(0) \leq \bar{s} \Rightarrow 0 \leq s(t) \leq \bar{s}, \forall t > t_0$, trivially from Equations (10), (11) and (12). Consider the storage function $W(\xi, \dot{\xi}, s) = \frac{1}{2} \dot{\xi}^T \mathbf{M} \dot{\xi} + \lambda_1 V_c(\xi) + s$, where $V_c(\xi)$ is the potential function associated with $\mathbf{f}_c(\xi)$. The rate of change of W is:

$$\dot{W}(\xi, \dot{\xi}) = \dot{\xi}^T \mathbf{M} \ddot{\xi} + \frac{1}{2} \dot{\xi}^T \dot{\mathbf{M}} \dot{\xi} + \lambda_1 \nabla V_c^T \dot{\xi} + \dot{s} \quad (15)$$

Substituting $\mathbf{M} \ddot{\xi}$ from Eq. (1) with τ_c given by Eq. (13) and using the skew-symmetry of $\mathbf{M} - 2\mathbf{C}$ as in Proposition 1 yields:

$$\begin{aligned} \dot{W}(\xi, \dot{\xi}) = & -\dot{\xi}^T \mathbf{D} \dot{\xi} + \dot{\xi}^T \tau_e + \beta_R(z, s) \lambda_1 \dot{z} + \\ & + \lambda_1 \dot{\xi}^T \mathbf{f}_c(\xi) + \lambda_1 \nabla V_c^T \dot{\xi} + \dot{s} \end{aligned} \quad (16)$$

The second-to-last two terms cancel because $\mathbf{f}_c(\xi) = -\nabla V_c(\xi)$. Substituting \dot{s} from Eq. (10) then yields:

$$\dot{W}(\xi, \dot{\xi}) = -\underbrace{\{1 - \alpha(s)\}}_{\geq 0} \dot{\xi}^T \mathbf{D} \dot{\xi} + \zeta(z, s) \lambda_1 \dot{z} + \dot{\xi}^T \tau_e \quad (17)$$

where $\zeta(z, s) = \beta_R(z, s) - \beta_s(z, s)$ has been introduced to ease the notation. Note that by Eq. (11) we have $1 - \alpha(s) \geq 0$. By Equations (12) and (14) we have that $\zeta(z, s) = 0$ for all $z > 0$ and $\zeta(z, s) \geq 0$ for $z < 0$. Hence, we have:

$$\dot{W}(\xi, \dot{\xi}) \leq \dot{\xi}^T \tau_e \quad (18)$$

which concludes the proof. \blacksquare

The specifications of the functions α, β_s, β_R allow some flexibility in the design. The functions used in this paper are described in the Appendix.

D. Decomposition of Task DS

The controller described in Section III-C relies on the decomposition of the DS into a conservative and a non-conservative

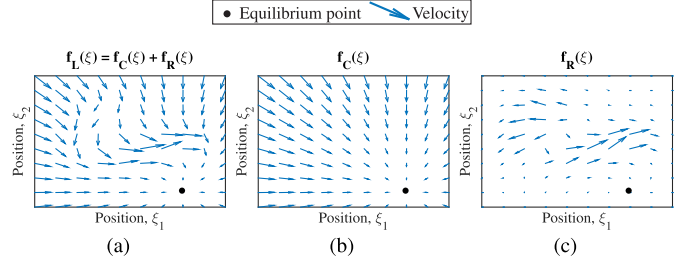


Fig. 2. The figures indicate the velocity from a DS evaluated over a 2-dimensional work-space. **a)** A task model represented with LMDS, Eq. (19). **b)** The conservative component (linear isotropic system used as original dynamics in LMDS), Eq. (20a) **c)** Non-conservative component, Eq. (20b).

part, as per Eq. (9). As shown in Section III-B, the conservative part of the DS can always be tracked. The non-conservative part may be scaled to zero if the energy tank is depleted, see Eq. (13). Importantly, the passivity of the controller does not rely on the decomposition of Eq. (9) being perfect (in other words, it is not necessary that \mathbf{f}_R is purely rotational). For example, any DS can be used with $\mathbf{f}_c = 0$ and $\mathbf{f}_R = \mathbf{f}$. The advantage of extracting a conservative component \mathbf{f}_c is that this component can always be followed even when the energy tank is depleted. Therefore, it is interesting to provide as good a decomposition as possible.

How to extract a conservative component from a DS depends on the method that was used to learn and encode the DS model. We recently proposed an approach for learning stable DS models based on some known *original dynamics*. Locally Modulated Dynamical Systems (LMDS) [6] is based on locally *reshaping* an existing model in a way that cannot alter the stability properties of the original model. Let $\mathbf{f}_L(\xi)$ denote an LMDS system and let $\mathbf{f}_O(\xi)$ denote the original dynamics. LMDS has the form:

$$\mathbf{f}_L(\xi) = \mathbf{G}(\xi) \mathbf{f}_O(\xi) \quad (19)$$

where $\mathbf{G}(\xi) \in \mathbb{R}^{N \times N}$ represents the continuous matrix valued *modulation function*. Reshaped systems result by designing the modulation function $\mathbf{G}(\xi)$ such that the desired behavior is achieved. In [6], we proposed an algorithm for incremental imitation learning in such systems. In that work, we also showed that highly complex systems can be modeled even if simple linear conservative systems are used as original dynamics. LMDS models \mathbf{f}_L that are based on conservative original dynamics \mathbf{f}_O have an implicit decomposition into conservative and non-conservative parts:

$$\mathbf{f}_c(\xi) = \mathbf{f}_O(\xi) \quad (20a)$$

and

$$\mathbf{f}_R(\xi) = (\mathbf{M}(\xi) - \mathbf{I}_{N \times N}) \mathbf{f}_O(\xi) \quad (20b)$$

This decomposition is illustrated in Fig. 2.

Several previously proposed methods for learning stable DS models in literature make use of a known Lyapunov function for ensuring stability of non-linear DS models during learning. Examples include DS learning with Extreme Learning Machines (ELM) [7] and our previous work Stable Estimator

of Dynamical Systems (SEDS) [5]. This class of DS have a straightforward decomposition given implicitly by the known Lyapunov function. Let $\mathbf{f}_P(\xi)$ denote such a system and let $V_P(\xi)$ denote the associated Lyapunov function. Since $V_P(\xi)$ is known, a conservative component can be found by taking its gradient:

$$\mathbf{f}_C(\xi) = -\nabla V_P(\xi) \quad (21a)$$

and

$$\mathbf{f}_R(\xi) = \mathbf{f}_P(\xi) - \mathbf{f}_C(\xi) = \mathbf{f}_P(\xi) + \nabla V_P(\xi) \quad (21b)$$

It is possible to combine the LMDS approach with batch-learning methods using a known Lyapunov function. For example, a stable SEDS model can serve as original dynamics in LMDS [6]. The conservative part of the SEDS model, Eq. (21a) would then also be the conservative part of the reshaped system, and the non-conservative part is simply the difference between the reshaped dynamics and the conservative part, see Eq. (9).

E. Impedance Adjustment

The architecture used in this paper differs fundamentally from the classical impedance control framework in that there is no notion of reference position. Instead, there is only a reference velocity, which is generated online as a function of the robot position. The classical mass-spring-damper model which is most commonly used in impedance control has the advantage that a designer can develop an intuition for how the system behavior will change as a result of a modification of the impedance parameters. This section is aimed at elucidating the link between the proposed controller and this model, to aid intuitive understanding of the behavior of the proposed system.

We write the closed loop dynamics of the system (1) under control $\tau_c = \mathbf{g}(\xi) - \mathbf{D}\dot{\xi} + \lambda_1 \mathbf{f}(\xi)$. Note that the passive controller derived in Section III-C yields the same closed loop behavior in the ideal case that the virtual storage is never depleted.

$$\mathbf{M}(\xi)\ddot{\xi} + (\mathbf{D}(\xi) + \mathbf{C}(\xi, \dot{\xi}))\dot{\xi} - \lambda_1 \mathbf{f}(\xi) = \tau_e. \quad (22)$$

Similarly to impedance controllers implemented without force sensing, it is not possible to alter the inertia of the system. We have control of the damping in directions orthogonal to the desired motion via the damping values $\lambda_2 \cdots \lambda_N$ which are allowed to vary with time, state or any other variable (see Section III-A). The stiffness term is replaced by $\lambda_1 \mathbf{f}(\xi)$ which can be interpreted as a nonlinear stiffness term. This interpretation is evident when considering the behavior of Eq. (22) close to the stable equilibrium point of \mathbf{f} . For simplicity, consider the robot in steady state ($\dot{\xi} = \ddot{\xi} = 0$) near the equilibrium point ξ^* such that $\mathbf{f}(\xi^*) = 0$. Accounting for steady state and approximating the left-hand side of Eq. (22) with a first order Taylor expansion around ξ^* then yields:

$$-\lambda_1 \left. \frac{\partial \mathbf{f}}{\partial \xi} \right|_{\xi=\xi^*} (\xi - \xi^*) = \tau_e \quad (23)$$

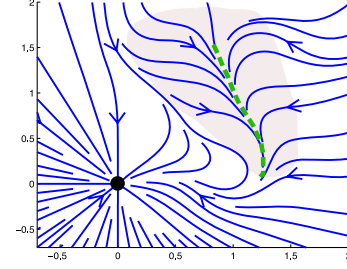


Fig. 3. A 2d illustration of a task model that locally encodes a stiffness-like behavior. The green dashed path corresponds to the reference trajectory. The shaded area roughly delineates the region in which the behavior of the DS is qualitatively similar to a stiffness attraction toward the reference trajectory. In this example, a perturbation from the nominal trajectory would yield a restoring force within the shaded region. The response to a large perturbation that makes the robot leave the shaded area is instead met with a completely different path to the target point.

corresponding to a steady-state stiffness equal to the Jacobian of \mathbf{f} at the equilibrium point scaled by the value of λ_1 . Globally, the term $\lambda_1 \mathbf{f}(\xi)$ can be interpreted as a nonlinear stiffness term centered on the equilibrium point of \mathbf{f} .

While the classical notion of stiffness manifests itself in vicinity of the equilibrium point of \mathbf{f} , it is not generally possible to generalize this to stiffness around general points in the work-space. To see this, consider again the steady-state linearization of the left-hand side of Eq. (22), but this time around an arbitrary point ξ' with $\mathbf{f}(\xi') \neq 0$:

$$-\lambda_1 \mathbf{f}(\xi') - \lambda_1 \left. \frac{\partial \mathbf{f}}{\partial \xi} \right|_{\xi=\xi'} (\xi - \xi') = \tau_e \quad (24)$$

A key observation is that a ‘stiffness behavior’ includes symmetry, where perturbations around a point on the desired trajectory are opposed uniformly around the reference trajectory. The DS task model, on the other hand, encodes infinitely many desired trajectories, given by the integral curves of \mathbf{f} . Hence, if the classical behavior of symmetrically converging toward a fixed trajectory is desired, this should be encoded in the task model \mathbf{f} . An example of a DS that locally encodes this spring-like behavior is given in Fig. 3.

IV. ROBOT EXPERIMENT

We consider a task of inserting a plate into a dish rack with perturbed location. A DS describing the task was learned by demonstration using LMDS [6], reshaping an original DS model encoded with SEDS [5] and extracted a conservative component from the DS as described in Section III-D. The motion is a parabolic reaching motion as depicted in Fig. 4.

A. Experimental Setup

The task DS has a single attractor to which all trajectories will converge. Ideally, this attractor would be placed exactly in the slot where the plate should be placed. In real scenarios, mismatch between environment state and the expected state is unavoidable. To account for this, we conducted three sets of task executions, in each of which the target location of the task DS

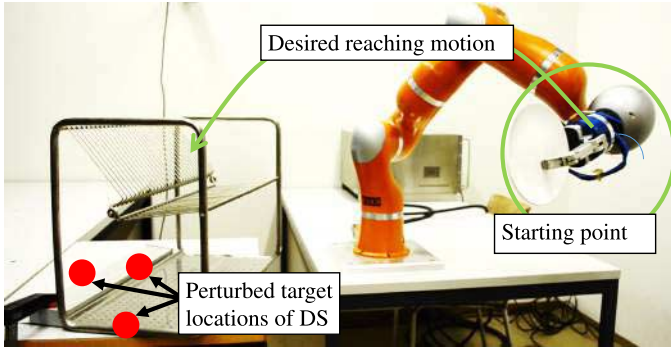


Fig. 4. The figure shows the experiment setup for the plate insertion task. The motion starts somewhere in the encircled region. The DS is describing a parabolic motion that has an attractor in one of three different perturbed target locations whose approximate position is shown with red dots.

was offset in different locations behind the real location of the dish rack, see Fig. 4. In each set of experiments, 5 task executions were carried out using two different controllers described below.

- A. The controller from Section III-C with $\bar{s} = s(0) = 10$, $\lambda_1 = 20$ and $\lambda_2 = \lambda_3 = 200$ is used. The value of λ_1 was chosen to the minimal value capable of overcoming static joint friction at the point of departure. The coupling functions are described in the Appendix.
- B. Trajectory integrated open-loop initialized by the starting position of the robot, tracked by a simple impedance controller without inertia shaping. The stiffness was set to $\mathbf{K} = k\mathbf{I}_{3 \times 3}$ with $k = 100$, the minimum value capable of reaching the final point of the task in free motion.

All task executions were started somewhere in a small region shown in Fig. 4. Rotational motion of the end-effector was in both cases simply damped by a high amount (4 Ns/rad) which effectively kept the orientation constant during the task execution.

B. Results

Since very low gains were used, and no inverse dynamics control was applied, it is not expected that either controller would be able to track the nominal motion given by \mathbf{f} with good accuracy. This is confirmed in Figures 6a, 6b and 6c which plot the nominal and actual trajectories for each setup for each perturbed location of the dish rack. Note especially the ‘shortcut’ tendency of the impedance controller. The proposed controller has a clear advantage in terms of respecting shape of the desired reaching motion. In this particular task, the shortcut effect meant that the robot was approaching the rack from the wrong direction, which sometimes lead to problematic final configurations as depicted in Figures 5a and 5b. In each of the three perturbed scenarios, controller A consistently placed the plate correctly because the pattern of approach was respected, see Fig. 5c. It should be pointed out that the degree to which the shape of the DS will be followed depends on the selection of the velocity feedback gain λ_1 but also on the damping gains λ_2 and λ_3 . In this task, we chose λ_2, λ_3 to relatively high values. This quickly dissipates kinetic energy in unwanted

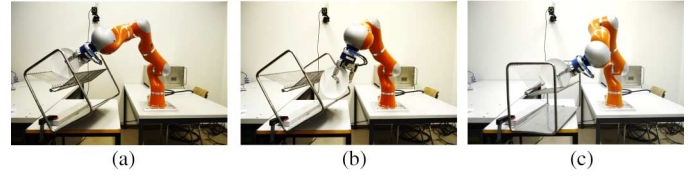


Fig. 5. **Left, Middle:** Examples of final configurations for controller B. **Right:** Typical final configuration for controller A.

directions and hence helps respect the shape of the motion as seen in Figures 6a, 6b and 6c, left plots.

As is clear from Figures 6d, 6e and 6f, controller A also has an advantage over controller B in terms of contact force after impact. At the time of impact, the reference point for controller B has already reached its final point, which is why there is no gradual ramp-up of the contact force as would normally be expected in contact with a timed trajectory. In the second perturbed location (Figures 6b and 6e) controller B resulted in some of the trials landing in a final configuration on the rack and some of the trials landed in a configuration under the rack. This is visible in the divergence of the trajectories near the end-point in Fig. 6b right, and also the high variance in the final contact force in Fig. 6e right. It should be emphasized that both controllers have been chosen to be as compliant as possible for this task, but the low stiffness is not enough to ensure a low contact force for positioning errors of this magnitude.

V. DISCUSSION

We have proposed a controller allowing to use the full modeling power of DS task representations at task execution time by closing the loop so that the DS is continuously updated with the actual state of the robot. We demonstrated the approach in a robotic manipulation task with unexpected collision and showed the advantage over the classical approach of integrating a reference trajectory and following it with an impedance controller. The controller is passive, guaranteeing stable interaction with any passive environment. We have exemplified it on DS learned using LMDS [6], but its applicability extends more generally to autonomous DS models such as SEDS [5] and ELM [7]. Possible applications for the proposed controller include **uncertain manipulation scenarios**, rehabilitation and Human-robot collaboration scenarios. The passivity property is necessary for safely interacting with humans although not sufficient. A robot controlled with PICDS can still be dangerous to a human depending on the DS model and/or velocity feedback gains.

There are works that have also made use of DS task description for ensuring passive control [12], [13]. These works propose **power-continuous controllers** that are interesting because no energy is dissipated and wasted. This is achieved with a virtual flywheel in [12] and with potential function [13]. In contrast to both [13] and [12] which both actively redirect any energy in the system along the desired DS, our approach uses dissipation as the main instrument to deal with perturbations. Note that a completely power-continuous controller will keep

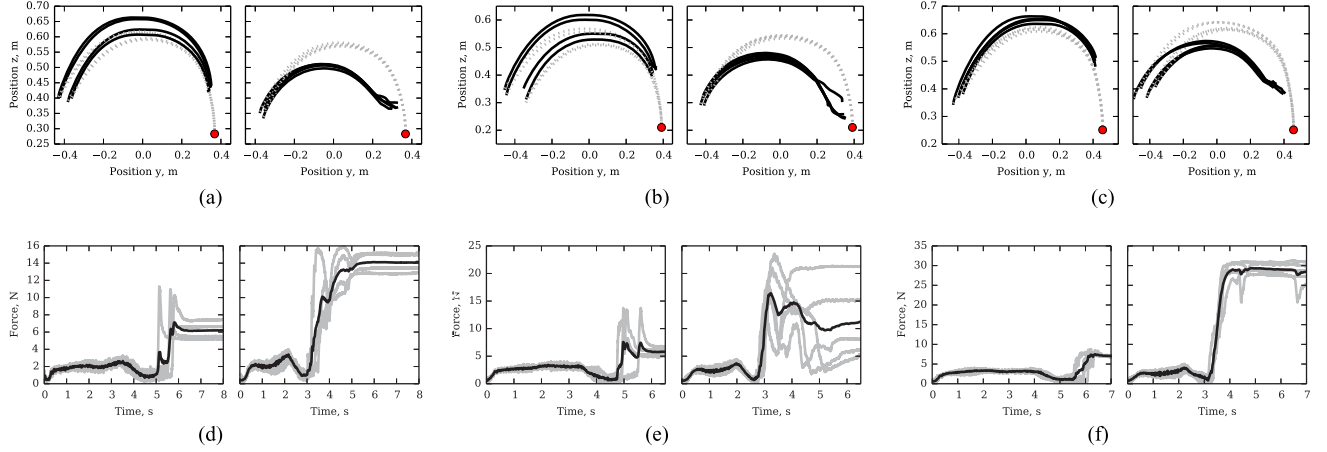


Fig. 6. **Top:** Actual (solid lines) and nominal (dotted lines) trajectories for the dish rack experiment. Figures **a,b,c** show the results of the three different perturbed scenarios, in each case with controller A on the left plot and controller B on the right plot. The motion in the YZ-plane is shown since the position in X direction remains almost constant during the motion. **Bottom:** The plots show the norm of the external force over time. The raw data are plotted in gray and a temporal average is plotted in black. Figures **d,e,f** show the data from each of the three perturbed scenarios with controller A on the left plot and controller B on the right plot. Due to imperfect estimation of the external force from the torque sensors of the robot, there is a small force even before impact with the dish rack.

any energy provided to it by the environment, hence the robot may accelerate sharply and change its direction of movement drastically if an strong external force is applied to it. It is easy to imagine scenarios where this may be counter-intuitive or even dangerous to a human working in proximity to the robot. Dissipation is a simpler, safer and more intuitive manner of dealing with undesired energy, albeit more wasteful.

As was pointed out in [14], the tuning of the mechanical impedance is not so clear in [12] and [13]. As an alternative solution, [14] proposed a method based on standard impedance control with a mechanism for regulating the speed along the desired trajectory to ensure passivity. The approach of [14] offers the possibility to passive tracking in contact, but lacks the flexibility of our approach since a single trajectory is used as reference.

The virtual storage concept used in Section III-C is not a novel idea but has been used by several works before us [19], [17]. Recent works making use of energy tanks are [20] which uses it to implement varying stiffness control, and [21] use the technique to incorporate sensor-based force tracking into Cartesian impedance control. Any method based on the concept of a limited virtual energy storage may not be able to control the system as desired, depending on external perturbations that are unknown prior to task execution. This is controlled via the choice of the upper bound for the energy storage, which is hence of great importance and should be chosen as a function of the task. This is also a potential problem of the PICDS controller proposed in this paper, which may be forced to fall-back to following the conservative part of the desired dynamics if the virtual storage is depleted. While this may be better than stopping completely, it will generally be detrimental to task performance and the starting level of the virtual storage hence needs to be chosen according to the task. This is especially critical if no conservative component can be extracted from the DS model. Future research will be aimed at how a suitable level of virtual storage can be learned under supervision of a human.

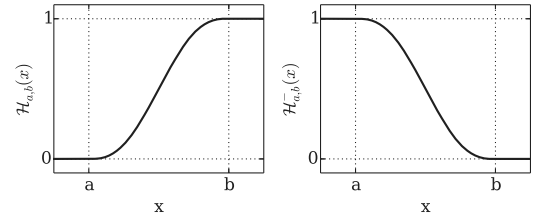


Fig. 7. The figure shows the smooth step functions used to compose α , β_s and β_R .

APPENDIX A SMOOTH COUPLING FUNCTIONS

In Section III-C, the scalar functions α , β_s and β_R were introduced. This appendix provides a one possible choice of these functions that satisfy Equations (11), (12) and (14) respectively.

Note that it is not necessary that the functions be continuous in z or s . However, to avoid sharp changes in control if the storage is depleted, we make use of smooth step functions for defining these coupling functions. To this end, we introduce the functions $\mathcal{H}_{a,b}$ and $\mathcal{H}_{a,b}^-$ denoting smooth step functions defined as:

$$\begin{cases} \mathcal{H}_{a,b}(x) = 0 & x < a \\ \mathcal{H}_{a,b}(x) = 6\left(\frac{x-a}{b-a}\right)^5 - 15\left(\frac{x-a}{b-a}\right)^4 + 10\left(\frac{x-a}{b-a}\right)^3 & a \leq x \leq b \\ \mathcal{H}_{a,b}(x) = 1 & x > b \end{cases} \quad (25a)$$

$$\mathcal{H}_{a,b}^-(x) = 1 - \mathcal{H}_{a,b}(x) \quad (25b)$$

The fifth order polynomial has the role of transitioning smoothly from 0 to 1 in the interval $[a, b]$, while matching first and second order derivatives ($= 0$) at $x = a$ and $x = b$. Fig. 7 illustrates these functions.

Starting by $\alpha(s)$, this function has the role of limiting the harvest of damping work which would otherwise be dissipated. This harvest (first term in Eq. (10), is only allowed as long as

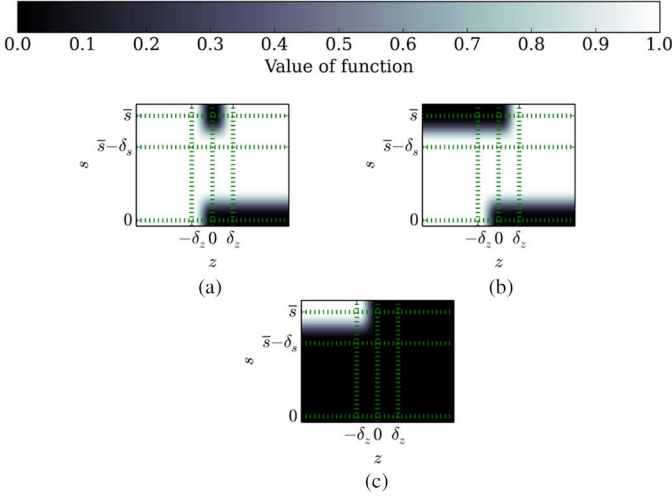


Fig. 8. The figure shows the β -functions and their difference over their 2d domain. Relatively high values of δ_s, δ_z are used for illustrations purposes. **a)** β_r **b)** β_s **c)** $\beta_r - \beta_s$.

there is still space in the virtual storage, i.e. $s < \bar{s}$. We define α simply as an inverted smooth step function:

$$\alpha(s) = \mathcal{H}_{\bar{s}-\delta_s, \bar{s}}^-(s) \quad (26)$$

where δ_s is a smoothness parameter determining the length of the transition from 1 to 0. In this work, $\delta_s = 0.1\bar{s}$ was always used.

The β_s function is defined as:

$$\beta_s(s, z) = 1 - \mathcal{H}_{-\delta_z, 0}(z) \mathcal{H}_{\bar{s}, \bar{s}+\delta_s}^-(s) - \mathcal{H}_{0, \delta_z}^-(z) \mathcal{H}_{\bar{s}-\delta_s, \bar{s}}^-(s) \quad (27)$$

where $\delta_z > 0$ is a smoothness parameter. The function is plotted in Fig. 8b.

Finally, the β_R function is defined as:

$$\beta_R(s, z) = \left(1 - \mathcal{H}_{-\delta_z, 0}(z) \mathcal{H}_{\bar{s}, \bar{s}+\delta_s}^-(s)\right) * \left(1 - \mathcal{H}_{-\delta_z, 0}(z) \mathcal{H}_{0, \delta_z}^-(z) \mathcal{H}_{\bar{s}-\delta_s, \bar{s}}^-(s)\right) \quad (28)$$

Fig. 8a plots β_R . To avoid unnecessarily scaling the non-conservative velocity component, δ_z should be chosen small. In this work, we have always used $\delta_z = 0.01$.

REFERENCES

- [1] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, pp. 1238–1274, 2013.
- [2] E. Gribovskaya, S. M. Khansari-Zadeh, and A. Billard, "Learning non-linear multivariate dynamics of motion in robotic manipulators," *Int. J. Robot. Res.*, vol. 30, no. 1, pp. 80–117, 2011.

- [3] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsargarakis, and D. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *Proc. 12th Int. Conf. Humanoid Robots*, 2012, pp. 323–329.
- [4] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2009, pp. 763–768.
- [5] S. M. Khansari-Zadeh and A. Billard, "Learning stable non-linear dynamical systems with Gaussian mixture models," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 943–957, Oct. 2011.
- [6] K. Kronander, S. M. Khansari-Zadeh, and A. Billard, "Incremental motion learning with locally modulated dynamical systems," *Robot. Auton. Syst.*, vol. 70, pp. 52–62, 2015.
- [7] A. Lemme, K. Neumann, R. F. Reinhart, and J. J. Steil, "Neural learning of vector fields for encoding stable dynamical systems," *Neurocomputing*, vol. 141, pp. 3–14, 2014.
- [8] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Auton. Robots*, vol. 32, no. 4, pp. 433–454, Mar. 2012.
- [9] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2009, pp. 2587–2592.
- [10] N. Hogan, "Impedance control: An approach to manipulation," *J. Dyn. Syst. Meas. Control*, vol. 107, no. 12, pp. 1–24, 1985.
- [11] E. Colgate and N. Hogan, "Robust control of dynamically interacting systems," *Int. J. Control*, vol. 48, no. 1, pp. 65–88, 1988.
- [12] P. Li and R. Horowitz, "Passive velocity field control of mechanical manipulators," *IEEE Trans. Robot. Autom.*, vol. 15, no. 4, pp. 751–763, Aug. 1999.
- [13] V. Duindam, S. Stramigioli, and J. Scherpen, "Passive compensation of nonlinear robot dynamics," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 480–487, Jun. 2004.
- [14] Y. Kishi, Z. Luo, F. Asano, and S. Hosoe, "Passive impedance control with time-varying impedance center," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, 2003, pp. 1207–1212.
- [15] S. M. Khansari-Zadeh and A. Billard, "Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions," *Robot. Auton. Syst.*, vol. 62, no. 6, pp. 752–765, Jun. 2014.
- [16] M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulators," *J. Dyn. Syst. Meas. Control*, vol. 103, pp. 119–125, 1981.
- [17] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. MacChelli, "Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 741–756, Aug. 2011.
- [18] D. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 354–369, Apr. 2010.
- [19] S. Stramigioli, C. Secchi, A. J. van der Schaft, and C. Fantuzzi, "Sampled data systems passivity and discrete port-Hamiltonian systems," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 574–587, Aug. 2005.
- [20] F. Ferraguti, C. Secchi, and C. Fantuzzi, "A tank-based approach to impedance control with variable stiffness," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2013, pp. 4948–4953.
- [21] C. Schindlbeck and S. Haddadin, "Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 440–447.