Autonomous Robots manuscript No.
(will be inserted by the editor)

Learning Potential Functions from Human Demonstrations with Encapsulated Dynamic and Compliant Behaviors

S.M. Khansari-Zadeh · O. Khatib

Received: date / Accepted: date

Abstract We consider the problem of devising a unified control policy capable of regulating both the robot motion and its physical interaction with the environment. We formulate this control policy by a non-parametric potential function and a dissipative field, which both can be learned from human demonstrations. We show that the robot motion and its stiffness behaviors can be encapsulated by the potential function's gradient and curvature, respectively. The dissipative field can also be used to model desired damping behavior throughout the motion, hence generating motions that follows the same velocity profile as the demonstrations. The proposed controller can be realized as a unification approach between "realtime motion generation" and "variable impedance control", with the advantages that it has guaranteed stability as well as does not rely on following a reference trajectory. Our approach, called Unified Motion and variable Impedance Control (UMIC), is completely time-invariant and can be learned from a few demonstrations via solving two (convex) constrained quadratic optimization problems. We validate UMIC on a library of 30 human handwriting motions and on a set of experiments on 7-DoF KUKA Light Weight Robot.

Keywords Potential field · variable impedance control · compliant control · robot learning · physical interaction control · motion control · imitation learning · motion primitives

1 Introduction

There is an increasing interest to move robots out of factory settings and introduce them into home environments.

Computer Science Department, Stanford University, USA
E-mail: {khansari;khatib}@cs.stanford.edu

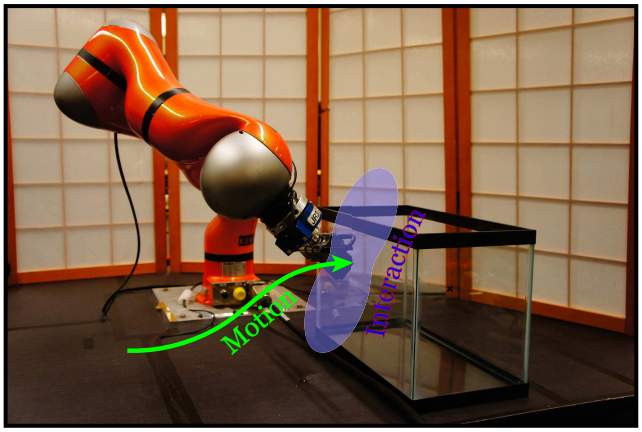


Fig. 1: Motion generation determines a path to reach the final state, whereas interaction control is concerned with the problem of describing the robot behavior when perturbed. Both of these skills are essential for safe and successful execution of many robotic tasks. In this graph, the belted-ellipsoid illustrates the robot stiffness at the contact point in Cartesian space between the end-effector and the glass.

Compared to strictly controlled environments, this new generation of robots should share a workspace with humans and cope with various sources of uncertainties while being able to perform a considerably wider set of motions. The classical way of controlling robots through a pure position controller falls short in addressing these challenges, and thus new techniques should be sought. Robots that are considered to work outside laboratories and factories should be able to safely and reliably interact with and explore the environment. Instead of precisely following a predefined path, they should adapt their motion in realtime to changes in dynamic environments and be inherently safe to operate if they collide with an unobserved object. They should also be endowed with some learning capability to increase their applicability beyond a repertoire of hard-coded motions.

For example, consider a task such as inserting a plug into a socket. In this task the robot first needs to know how

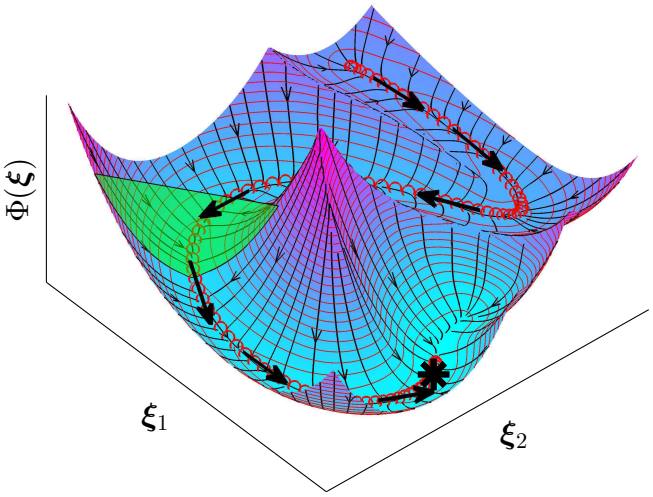


Fig. 2: Our approach unifies motion generation and variable impedance control by devising a potential function $\Phi(\boldsymbol{\xi})$, $\boldsymbol{\xi} \in \mathbb{R}^d$ that captures this information. The energy levels and energy gradients are shown in solid red and solid black lines, respectively. The potential energy Φ is learned from demonstrations (shown in red circles), and can be pictured as a valley with two significant parameters: a slope ϱ and a curvature κ. The slope ϱ captures the motion behavior. If we drop a virtual ball on any of the demonstration points, this virtual ball follows the rest of the demonstration with an acceleration governed by ϱ and stops at the target point (illustrated in black star). The arrows highlights the motion due to the slope ϱ. The curvature κ is perpendicular to the desired direction of motion and encodes the stiffness behavior. The green surface visualizes κ at a demonstration data point. The higher the κ the more resistance the virtual ball shows to perturbations, hence returning faster to the nominal motion. Though not illustrated in this figure, our method also encodes the damping parameter throughout the state space to avoid oscillation. Note that both ϱ and κ are state dependent parameters and could vary throughout the motion.

to move its links to reach the power strip not only from one particular position, but from any point in the space. Motion generators such as potential fields (Khatib, 1986), dynamical systems (Khansari-Zadeh and Billard, 2011), or sample-based planners (Kavraki and LaValle, 2008) could efficiently provide this path when the environment is completely known. Due to uncertainties in estimating the socket position as well as the robot model, it is very unlikely that the plug perfectly aligns with the socket holes and thus will not be inserted properly. Typical situations that may be encountered are: the plug does not touch the power strip, it collides with the surface of the power strip earlier than expected, only one of the rods is inside its corresponding hole, and so on. Even if aligned, inserting the plug into the hole with a pure motion generator is challenging because slight uncertainty could result in a large force and eventually lead to instability and breakage. This is one of many examples in which a pure motion generation is prone to failure. This task as well as many other tasks that a robot may face in human environments emphasizes the necessity of controlling both the robot motion and its physical interaction with the environment, see Fig. 1 (Hogan, 1985; Khatib, 1987).

The term "interaction control" refers to regulation of the robot's dynamic behavior at its contact points/surfaces, at which energy may be exchanged with the environment (Hogan and Buerger, 2005). Interaction control is a fundamental requirement for successful execution of a task when it includes contact with an environment (e.g consider tasks such as polishing, assembly, deburring, transferring, machining, and so on). It also plays an important role in unstructured environments, where exact localization of objects is not possible, and the robot should safely and compliantly touch the environment in order to perform the task (Khatib et al., 2008; Villani and Schutter, 2008).

Variable impedance control is one promising solution to control the robot physical interaction with the environment through tunable impedance parameters. It allows customizing of the robot interaction property while the task is being executed, for instance making the robot compliant in uncertain regions, but stiff when necessary. The successful implementation of variable impedance control has been shown in several works (Ferraguti et al., 2013; Ganesh et al., 2012; Kronander and Billard, 2013), where a time-indexed trajectory is tracked, with the robot's impedance properties changing along the trajectory.

To provide realtime adaptation to changing environments (e.g. if the target point or other objects close to the robot are displaced), a variable impedance controller should also be endowed with a realtime motion generation capability. However, there are two challenges in achieving this goal: 1) ensuring stability of variable impedance control even for following a time-indexed trajectory is difficult, and the addition of realtime motion generator significantly aggravates it, and 2) the realtime motion generator operates disjointedly from the variable impedance controller and is thus unaware of its limitations. Hence it may generate trajectories that are infeasible or non-optimal for the variable impedance controller.

To summarize, robots that are considered to work in human environments should be endowed with three key skills: 1) the ability to adapt their motion in realtime to changing environments, 2) the skill to smoothly shift from compliant to stiff (and vice-versa) when necessary, and 3) the learning capability to extend the robot applicability beyond a set of hard-coded motions. In this paper, we propose a novel approach that can provide these three skills in performing episodic tasks. We achieve this goal by proposing a non-parametric potential function that can capture both motion and impedance properties (see Fig. 2). Our proposed controller, called Unified Motion and variable Impedance Control (UMIC), allows the robot to safely and accurately operate in a human environments. An estimate of UMIC

can be learned from one or more demonstrations by solving two convex constrained optimization problems. UMIC is time-invariant[1] and is guaranteed to be stable when it comes into contact with passive environments. We validate the performance of our controller in simulation and on the 7-DoF KUKA Light Weight Robot (LWR).

The remainder of this paper is structured as follows. Section 2 describes our proposed non-parametric potential function. Section 3 introduces the UMIC control policy and Section 4 describes how it can be learned from demonstrations by solving two quadratic optimization problems. Section 5 focuses on stability analysis of the proposed controller. Section 6 reports on experimental evaluation on simulated and real robots. Section 7 discusses different aspects of the proposed method and compares it with two state-of-the-art variable impedance control techniques. Section 8 reviews the related work, and finally Section 9 summarizes and concludes the paper.

Note that to enhance the readability of equations we use the following convention throughout the paper: typeface to refer to scalars (e.g. a), lowercase bold font for vectors (e.g. $\boldsymbol{a}$), and uppercase bold font for matrices (e.g. $\boldsymbol{A}$). Unless otherwise specified, we consider the following notations with their unit specified in parenthesis:

- Potential energy terms Φ and $\phi_0(Joule)$.
- When referred to linear motions: state variable $\boldsymbol{\xi}(m)$, stiffness $\boldsymbol{S}(N/m)$, damping $\boldsymbol{D}(N.s/m)$, force $\boldsymbol{f}(N)$, and dissipative field $\boldsymbol{\Psi}(N)$.
- When referred to angular motions: state variable $\boldsymbol{\xi}(rad)$, stiffness $\boldsymbol{S}(N.m/rad)$, damping $\boldsymbol{D}(N.m.s/rad)$, torque $\boldsymbol{f}(N.m)$, and dissipative field $\boldsymbol{\Psi}(N.m)$.

Note that our state variables can be composed of both linear and angular motions, which in this case we have a combination of these units. When describing our method through examples, without loss of generality, we consider a state variable that is defined in Cartesian space with the following structure: $\boldsymbol{\xi} = [\boldsymbol{\xi}_1 \;\; \boldsymbol{\xi}_2 \;\; \boldsymbol{\xi}_3]^T = [x \;\; y \;\; z]^T$.

2 Formalism

Consider a state variable $\boldsymbol{\xi} \in \mathbb{R}^d$ that can be used to unambiguously define the state of a robotic system. The state variable $\boldsymbol{\xi}$, for instance, could represent the robot's generalized joint angles, the position and orientation of the end-effector, or solely position or orientation of the end-effector. We define our control policy $\boldsymbol{\tau}_c \in \mathbb{R}^d$ as the negative gradient of a scalar time-invariant potential function $\Phi(\boldsymbol{\xi}) : \mathbb{R}^d \mapsto \mathbb{R}^+$ minus a dissipative field $\boldsymbol{\Psi}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) : \mathbb{R}^{d\times d} \mapsto \mathbb{R}^d$:

$$\boldsymbol{\tau}_c = -\nabla\Phi(\boldsymbol{\xi}) - \boldsymbol{\Psi}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) \tag{1}$$

[1] A time-invariant system is a system whose output does not explicitly depend on time. Note that such system could have dependency on time derivatives of the state variable (Slotine and Li, 1991).

Note that if the state variable $\boldsymbol{\xi}$ is defined as the generalized joint angles, $\boldsymbol{\tau}_c$ directly corresponds to the actual torque command that should be sent to the actuators. In contrast, if $\boldsymbol{\xi}$ is defined in task space, we use operational space formulation (Khatib, 1987) to compute the actuators torque command from $\boldsymbol{\tau}_c$.

The main advantage of defining $\boldsymbol{\tau}_c$ as described above is that once a potential function is determined for the task at hand, stability of the controller unfolds naturally through the use of Lyapunov's direct method. If we determine a rich representation for $\Phi(\boldsymbol{\xi})$ and $\boldsymbol{\Psi}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})$ that can convey information about both motion and impedance properties of the task, then we reduce the complex problem of 'ensuring stability of a variable impedance controller with feedback motion planning' to a much simpler problem of learning a potential function $\Phi(\boldsymbol{\xi})$ and a dissipative field $\boldsymbol{\Psi}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})$.

2.1 Data Collection

Let us assume we are provided with a set of N kinesthetic demonstrations $\{\boldsymbol{\xi}^{t,n}, \dot{\boldsymbol{\xi}}^{t,n}, \boldsymbol{\tau}^{t,n}\}_{t=0,n=1}^{T^n,N}$ and their corresponding stiffness property $\{\boldsymbol{S}^{t,n}\}_{t=0,n=1}^{T^n,N}$, where $\boldsymbol{S}^{t,n} \in \mathbb{R}^{d\times d}$ are positive definite matrices. Without loss of generality, we assume the task is defined in the target frame of reference, i.e. $\boldsymbol{\xi}^{T^n,n} = \boldsymbol{\xi}^* = \boldsymbol{0}, \forall n \in 1..N$. This can be achieved by a simple translation of demonstrations. Furthermore, to avoid presence of several indices, we simplify the notation by concatenating all the demonstrations for each variable into one single vector. Thus, instead of referring to the demonstrations as $\{(.)^{t,n}\}_{t=0,n=1}^{T^n,N}$, we use the notation $\{(.)^i\}_{i=1}^{\mathcal{T}}$, where $\mathcal{T} = \sum_{n=1}^{N} T^n$ is the total number of datapoints. The index i can be easily computed for each (t, n) with a simple algorithm. To avoid addressing the correspondence problem, demonstration trajectories are shown from the robot's point of view, by the user guiding the robot's arm passively through the task (i.e. kinesthetic demonstration). Note that throughout this paper we use the term *centers* to refer to $\boldsymbol{\xi}^i$, while we use *datapoints* to refer to the whole collected data set, i.e. including $\boldsymbol{\xi}^i$, $\dot{\boldsymbol{\xi}}^i$, $\boldsymbol{\tau}^i$, and $\boldsymbol{S}^i$.

Stiffness property can also be collected from different strategies such as: direct mapping from pressure sensor mounted on the wrist of the robot (Kronander and Billard, 2013), being inversely proportional to the variance of the demonstrations (Calinon et al., 2010b), being extracted from human demonstrations (Howard et al., 2013), or manually defined as a (task-dependent) function of other variables (this paper, see Section 6 for more detail). Note that extracting stiffness and damping property from human demonstrations is not the main focus of this paper. In this paper we assume this information is available using one of the state of the art techniques. We refer interested readers to (Howard et al., 2013; Kronander and Billard, 2013) for more information on collection of impedance properties from human.

2.2 Potential Energy

We associate an energy element $\phi^i : \mathbb{R}^d \mapsto \mathbb{R}^+$ to each of the demonstration data points $\boldsymbol{\xi}^i$:

$$\phi^i(\boldsymbol{\xi}) = \phi_0^i + \frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\xi}^i)^T \boldsymbol{S}^i(\boldsymbol{\xi} - \boldsymbol{\xi}^i) \qquad \forall i \in 1..\mathcal{T} \quad (2)$$

where $\phi_0^i \in \mathbb{R}^+$ is a constant scalar, and $(.)^T$ denotes the transpose. For each energy element $\phi^i(\boldsymbol{\xi})$, the force by which a particle $\boldsymbol{\xi}$ is attracted to the center $\boldsymbol{\xi}^i$ is given by $-\boldsymbol{S}^i(\boldsymbol{\xi} - \boldsymbol{\xi}^i)$. Thus the higher the $\boldsymbol{S}^i$, the more the attraction force is. Although the constant ϕ_0^i seems unnecessary when we solely look at one energy element, as we will illustrate in Section 3, it is of significant importance when we consider the accumulating effect of all energy elements.

We choose the kernel regression method to build the total energy (potential) function based on the energy elements $\phi^i(\boldsymbol{\xi})$. At a query point $\boldsymbol{\xi} \in \mathbb{R}^d$, we determine the contribution of each energy element using Gaussian kernel:

$$\omega^i(\boldsymbol{\xi}) = e^{-\frac{1}{2(\sigma^i)^2}(\boldsymbol{\xi}-\boldsymbol{\xi}^i)^T(\boldsymbol{\xi}-\boldsymbol{\xi}^i)} \qquad \forall i \in 1..\mathcal{T} \quad (3)$$

where $\sigma^i \in \mathbb{R}^+$ are smoothing parameters controlling the region of influence of each energy element. The total potential energy at $\boldsymbol{\xi}$ is given by:

$$\Phi(\boldsymbol{\xi}) = \frac{\sum_{i=1}^{\mathcal{T}} \omega^i(\boldsymbol{\xi})\phi^i(\boldsymbol{\xi})}{\sum_{j=1}^{\mathcal{T}} \omega^j(\boldsymbol{\xi})} \quad (4)$$

Let us simplify the notation by denoting $\sum_{i=1}^{\mathcal{T}}$ with $\sum_i$ and defining the normalized weights $\tilde{\omega}^i(\boldsymbol{\xi})$ by:

$$\tilde{\omega}^i(\boldsymbol{\xi}) = \frac{\omega^i(\boldsymbol{\xi})}{\sum_j \omega^j(\boldsymbol{\xi})} \qquad \forall i \in 1..\mathcal{T} \quad (5)$$

Then Eq. (4) can be simplified (notation-wise) to:

$$\Phi(\boldsymbol{\xi}) = \sum_i \tilde{\omega}^i(\boldsymbol{\xi})\phi^i(\boldsymbol{\xi}) \quad (6)$$

Note that the nonlinear weights $\tilde{\omega}^i(\boldsymbol{\xi})$ have the following two properties: $0 < \tilde{\omega}^i(\boldsymbol{\xi}) \leq 1$ and $\sum_i \tilde{\omega}^i(\boldsymbol{\xi}) = 1$, $\forall \boldsymbol{\xi} \in \mathbb{R}^d$. Considering these properties and the fact that $\phi^i(\boldsymbol{\xi})$ are positive scalars (see Eq. (2)) yields the relation $\Phi(\boldsymbol{\xi}) \geq 0$, $\forall \boldsymbol{\xi} \in \mathbb{R}^d$.

Figure 3 shows potential energy elements for a one dimensional trajectory defined by three points. In regions close to a data point, its associated weight increases, and thus the total energy smoothly switches to the potential energy element of that data point (which takes a quadratic form). Depending on the values of $\boldsymbol{\xi}^i$, $\boldsymbol{S}^i$, ϕ_0^i, and σ^i, the potential energy function may have one or more minima. Since the centers $\boldsymbol{\xi}^i$ and their associated stiffness $\boldsymbol{S}^i$ are given by the user, we need to determine the value of ϕ_0^i and σ^i so as to force $\Phi(\boldsymbol{\xi})$ to have one single minimum, which should be

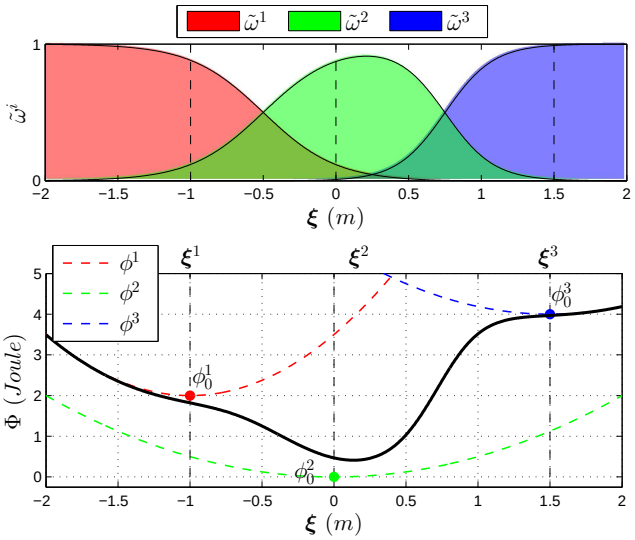


Fig. 3: Illustration of the potential energy for a 1 dimensional trajectory defined by three data points. The total potential energy (shown in black line) is determined from the nonlinear weighted sum of potential energy elements (shown in colored dashed lines). Nonlinear weights associated to each energy element is illustrated on the top. In regions close to a data point, its associated weight increases, and thus the total energy smoothly switch to the potential energy element of that data point. The following values are used: $\boldsymbol{\xi}^1 = -1$, $\boldsymbol{\xi}^2 = 0$, $\boldsymbol{\xi}^3 = 1.5$, $\phi_0^1 = 2$, $\phi_0^2 = 0$, $\phi_0^3 = 4$, $\boldsymbol{S}_0^1 = 3$, $\boldsymbol{S}_0^2 = 1$, $\boldsymbol{S}_0^3 = 1.5$, and $\sigma^1 = \sigma^2 = \sigma^3 = 0.5$.

centered at the desired final point $\boldsymbol{\xi}^*$. In the example shown in Fig. 3, $\Phi(\boldsymbol{\xi})$ has its global minimum at $\boldsymbol{\xi} = 0.136$. To force the minima to occur at $\boldsymbol{\xi}^* = 0$, one solution could be to change ϕ_0^3 from 4 to 16.2. We will elaborate on how we learn the values of ϕ_0^i in Section 4.

2.3 Dissipative Field

We associate a simple linear dissipative field $\boldsymbol{\psi}^i(\dot{\boldsymbol{\xi}}) : \mathbb{R}^d \mapsto \mathbb{R}^d$ to each data point, which is given by

$$\boldsymbol{\psi}^i(\dot{\boldsymbol{\xi}}) = \boldsymbol{D}^i\dot{\boldsymbol{\xi}} \quad (7)$$

where $\boldsymbol{D}^i \in \mathbb{R}^{d\times d}$ are positive definite matrices. The total dissipative energy can be computed through nonlinear weighted sum of each dissipative element $\boldsymbol{\psi}^i(\dot{\boldsymbol{\xi}})$:

$$\boldsymbol{\Psi}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \sum_i \tilde{\omega}^i(\boldsymbol{\xi})\boldsymbol{\psi}^i(\dot{\boldsymbol{\xi}}) \quad (8)$$

3 Unified Motion and Impedance Controller

Our Unified Motion and Impedance Control policy can be obtained by taking the gradient of Eq. (6) and substituting it as well as Eq. (8) into Eq. (1). After doing a few rearrangements, we obtain:

$$\boldsymbol{\tau}_c = \sum_i \frac{1}{(\sigma^i)^2}\tilde{\omega}^i(\boldsymbol{\xi})\Big(\phi^i(\boldsymbol{\xi}) - \Phi(\boldsymbol{\xi})\Big)(\boldsymbol{\xi} - \boldsymbol{\xi}^i) - \tilde{\omega}^i(\boldsymbol{\xi})\Big(\boldsymbol{S}^i(\boldsymbol{\xi} - \boldsymbol{\xi}^i) + \boldsymbol{D}^i\dot{\boldsymbol{\xi}}\Big) \quad (9)$$

There are three main terms in Eq. (9). Let us define:

$$\boldsymbol{\tau}^i_{nominal} = \frac{1}{(\sigma^i)^2}\tilde{\omega}^i(\boldsymbol{\xi})\Big(\phi^i(\boldsymbol{\xi}) - \Phi(\boldsymbol{\xi})\Big)(\boldsymbol{\xi} - \boldsymbol{\xi}^i) \tag{10a}$$

$$\boldsymbol{\tau}^i_{attract} = -\tilde{\omega}^i(\boldsymbol{\xi})\boldsymbol{S}^i(\boldsymbol{\xi} - \boldsymbol{\xi}^i) \tag{10b}$$

$$\boldsymbol{\tau}^i_{damp} = -\tilde{\omega}^i(\boldsymbol{\xi})\boldsymbol{D}^i\dot{\boldsymbol{\xi}} \tag{10c}$$

then we have:

$$\boldsymbol{\tau}_c = \sum_i \boldsymbol{\tau}^i_{nominal} + \sum_i \boldsymbol{\tau}^i_{attract} + \sum_i \boldsymbol{\tau}^i_{damp} \tag{11a}$$

$$= \boldsymbol{\tau}_{nominal} + \boldsymbol{\tau}_{attract} + \boldsymbol{\tau}_{damp} \tag{11b}$$

Let us introduce the following two definitions before further discussion:

Direction of nominal motion: At a center point $\boldsymbol{\xi}^i$, the direction of nominal motion is the direction that connects a center $\boldsymbol{\xi}^i$ to its next center $\boldsymbol{\xi}^{i+1}$ in the demonstration. In fact, if the robot starts the motion from $\boldsymbol{\xi} = \boldsymbol{\xi}^1$ (i.e. top-center of the graph in Fig. 2), we ideally want it to go through all the centers and stop at $\boldsymbol{\xi}^{\mathcal{T}}$. The terms $\boldsymbol{\tau}^i_{nominal}$ in Eq. (11) are mainly responsible to generate the nominal motion. In Fig. 2, the nominal direction of motion is indicated by arrows.

Direction of attraction: At a point $\boldsymbol{\xi} \in \mathbb{R}^d$, the direction of attraction is a vector that connects $\boldsymbol{\xi}$ to the closest center point $\boldsymbol{\xi}^i$ in the demonstrations. For example in Fig. 2, the direction of attraction lies on the green vertical surface for any point $\boldsymbol{\xi}$ on the surface. The terms $\boldsymbol{\tau}^i_{attract}$ in Eq. (11) are mainly responsible to generate the attraction force towards the nominal motion.

Figure 4 provides an illustration of Eq. (9). For clarity, only three center-points are shown. The query point $\boldsymbol{\xi}$ is indicated by a hollow circle. The nonlinear weights $\tilde{\omega}^i(\boldsymbol{\xi})$, $\tilde{\omega}^{i+1}(\boldsymbol{\xi})$, and $\tilde{\omega}^{i+2}(\boldsymbol{\xi})$ are respectively shown in red, green, and blue through an RGB color map. For instance, the more red the color is at a region, the higher the value $\tilde{\omega}^i(\boldsymbol{\xi})$ has at that region. At the query point, we could observe that the center point $\boldsymbol{\xi}^{i+1}$ is more dominant than the other two terms. The terms $\boldsymbol{\tau}^j_{nominal}$, $j = i..i+2$ are illustrated as a triangular-headed arrows for each center. These terms determines how the motion should evolve to create a behavior similar to the nominal motion. For example, if the robot is currently at $\boldsymbol{\xi} = \boldsymbol{\xi}^i$, these terms derives the robot towards the next data point in the demonstration, which is $\boldsymbol{\xi}^{i+1}$. The length of the vector indicates its magnitude. As it is expected, it is larger for $\boldsymbol{\xi}^{i+1}$ due to the higher value of $\tilde{\omega}^{i+1}(\boldsymbol{\xi})$.

The sharp-headed arrows illustrate the terms $\boldsymbol{\tau}^j_{attract}$, $j = i..i+2$, which define the attraction force towards the nominal motion. These terms can be perceived as a set of springs that connect the query point to each center points. Due to the nonlinear weights, the effect of these springs varies based on the query point $\boldsymbol{\xi}$. The diamond-headed arrows indicate the terms $\boldsymbol{\tau}^j_{damp}$, $j = i..i+2$, which act as dissipating elements; pumping the energy out of the system.

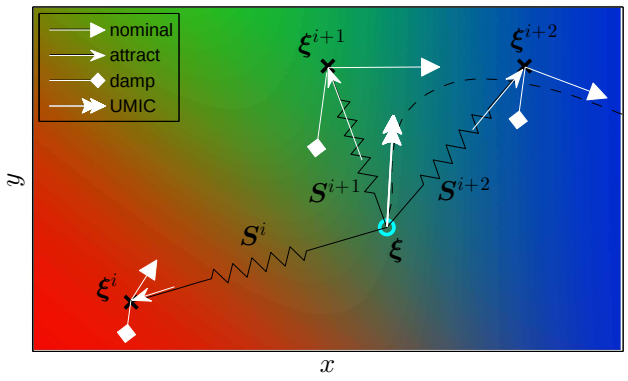


Fig. 4: Illustration of UMIC as per Eqs. (9) and (11). In this graph we consider $\boldsymbol{\xi} = [x \;\; y]^T$. For clarity, only three center-points are considered (indicated by a cross). The query point $\boldsymbol{\xi}$ is indicated by a hollow circle. The nonlinear weights $\tilde{\omega}^i(\boldsymbol{\xi})$, $\tilde{\omega}^{i+1}(\boldsymbol{\xi})$, and $\tilde{\omega}^{i+1}(\boldsymbol{\xi})$ are respectively shown in red, green, and blue through an RGB color map. For instance, the more red the color is at a region, the higher the value $\tilde{\omega}^i(\boldsymbol{\xi})$ has at that region. The UMIC control policy, i.e. $\boldsymbol{\tau}_c$ in Eq. (9), is illustrated by a double-headed arrows. The labels nominal, attract, and damp are defined according to Eq. (11). The dashed line shows the resulting path by integrating the motion from the query point $\boldsymbol{\xi}$. For further details refer to Section 3.

The UMIC control policy $\boldsymbol{\tau}_c$, i.e. the net effect of all these forces, is shown by a double-headed arrow. This value is mostly influenced by the center $\boldsymbol{\xi}^{i+1}$ at the current query point. This term pushes the motion towards $\boldsymbol{\xi}^{i+1}$ and slightly to its right. If we continue the motion for a few iterations, the query point will pass near $\boldsymbol{\xi}^{i+2}$ and eventually converges to the nominal motion (see the dashed line). Note that all the vectors in Fig. 4 are essentially acting at the query point; however for the clarity of the graph, we draw each of them at its corresponding center point.

As stated before, each term in Eq. (9) is dominant in providing a certain behavior. The term $\boldsymbol{\tau}^i_{nominal}$ defines the nominal motion. Although it does not explicitly appear in Eq. (9), through a few rearrangements we could observe that ϕ^i_0 in Eq. (2) play a major role in changing the value of $\boldsymbol{\tau}_{nominal}$:

$$\begin{aligned}\boldsymbol{\tau}_{nominal} &= \sum_i \frac{1}{(\sigma^i)^2}\tilde{\omega}^i(\boldsymbol{\xi})\Big(\phi^i(\boldsymbol{\xi}) - \Phi(\boldsymbol{\xi})\Big)(\boldsymbol{\xi} - \boldsymbol{\xi}^i) \\ &= \sum_i \frac{\tilde{\omega}^i(\boldsymbol{\xi})}{(\sigma^i)^2}(\boldsymbol{\xi} - \boldsymbol{\xi}^i)\Big(\phi^i_0 - \sum_j \tilde{\omega}^j(\boldsymbol{\xi})\phi^j_0 + \cdots \\ &\quad + \upsilon^i(\boldsymbol{\xi}; \boldsymbol{S}^i, \boldsymbol{\xi}^i) - \sum_j \tilde{\omega}^j(\boldsymbol{\xi})\upsilon^j(\boldsymbol{\xi}; \boldsymbol{S}^j, \boldsymbol{\xi}^j)\Big)\end{aligned} \tag{12}$$

where $\upsilon^i(\boldsymbol{\xi}; \boldsymbol{S}^i, \boldsymbol{\xi}^i) = \frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\xi}^i)^T\boldsymbol{S}^i(\boldsymbol{\xi} - \boldsymbol{\xi}^i)$ is a quadratic function. As can be seen in Eq. (12), $\boldsymbol{\tau}_{nominal}$ is a linear function of ϕ^i_0. Furthermore, by inspecting Eqs. (10b) and (10c) we could observe that none of the terms $\boldsymbol{\tau}_{attract}$ and $\boldsymbol{\tau}_{damp}$ depends on ϕ^i_0. We will use these two properties later on

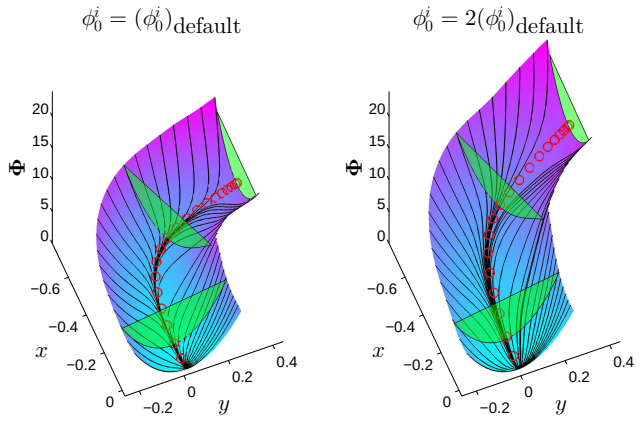


Fig. 5: Effect of ϕ_0^i on the potential energy gradient. The values of ϕ_0^i in the right figure are twice the value of ϕ_0^i in the left graph. All other parameters are identical between the two graphs. Demonstrations are shown by red circles. Increasing the value of ϕ_0^i results in increasing the value of $\nabla\Phi$ along the direction of motion, while the attraction force towards the nominal motion remains unchanged (compare curvature of corresponding green surfaces). Note that to enhance clarity, the potential energy is only illustrated on a portion of x-y domain.

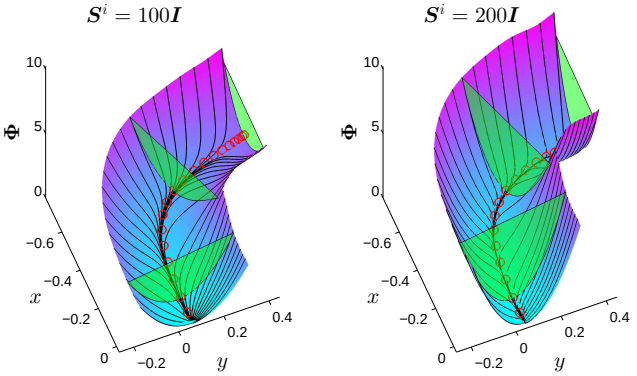


Fig. 6: Effect of stiffness on the potential energy gradient. The stiffness values on the right graph are twice the ones on the left. All other parameters are identical between the two graphs. The motion starts from top-right of each graph and terminates at the bottom. Demonstrations are shown by red circles. By increasing stiffness, the valley in the potential function becomes steeper, hence points are attracted towards the centers more rapidly. In contrast the slope along the nominal motion is identical between the two graphs. This is due to the fact that this slope is mainly governed by the parameter ϕ_0^i. Note that to enhance clarity, the potential energy is only illustrated on a portion of x-y domain.

to learn ϕ_0^i from demonstrations by solving a convex constrained optimization problem. Also note that $\boldsymbol{\tau}_{nominal}$ linearly depends on the stiffness matrices $\boldsymbol{S}^j$.

Figure 5 illustrates the effect of changing ϕ_0^i on $\boldsymbol{\tau}_{nominal}$. In this graph, the motion starts from the top-right and terminates at the bottom of the graph. The green parabolic-shaped surface illustrates the curvature of the potential function at one of the center points $\boldsymbol{\xi}^i$. The values of ϕ_0^i on the right graph are twice the value of ϕ_0^i in the left graph. All the other parameters are identical between the two graphs. As can be seen the curvature of the potential function, highlighted by green surfaces, is (nearly) identical in both graphs. This indicates that the attracting force towards the nominal motion is the same in both cases. However, each center point at the model on the right achieves a higher potential energy than its corresponding center on the left. As a result, the model on the right generates a greater force along the nominal motion than the one on the left.

The terms $\boldsymbol{\tau}_{attract}^i$ mostly determine the impedance property, i.e. how much the robot should resist when perturbed along the direction orthogonal to the nominal motion. From a different perspective, if the robot state $\boldsymbol{\xi}$ is within the region of influence of a center $\boldsymbol{\xi}^i$, the higher $\boldsymbol{S}^i$, the faster the robot moves towards $\boldsymbol{\xi}^i$. Figure 6 illustrates the effect of changing stiffness on the control policy. As highlighted before, first observe that the value of stiffness can be visually perceived as the curvature of the potential field. The higher the stiffness $\boldsymbol{S}^i$, the more curvature the potential field has at $\boldsymbol{\xi}^i$. As can be seen, the change in stiffness mainly affects the component of $\boldsymbol{\tau}_c$ along the direction of attraction. The centers on both models have nearly identical potential energy, therefore generating similar forces along the nominal motion.

Note that all the computations to determine the control command $\boldsymbol{\tau}_c$ are performed online. The computational complexity of our controller grows linearly with the number of data points $\mathcal{T}$ and quadratically with the dimensionality d. If diagonal stiffness and damping are used, then computational complexity grows linearly in both $\mathcal{T}$ and d. As a result, in either case, the required number of operations to compute $\boldsymbol{\tau}_c$ is tractable and as we showcase through several robot experiments, online computation of the control command can satisfy the strict realtime requirement enforced by robots such as KUKA Light Weight Robot.

4 Learning the UMIC Control Policy

In Section 3, we presented the UMIC control policy. It remains now to determine a procedure to build an estimate of the UMIC from a set of user demonstrations. The centers $\boldsymbol{\xi}^i$ and their associated stiffness $\boldsymbol{S}^i$ are known and given by the collected demonstrations. It now remains to determine the value of σ^i, ϕ_0^i, and $\boldsymbol{D}^i$ such that the target point becomes the attractor of the potential function $\Phi(\boldsymbol{\xi})$ and the robot follows the same velocity-profile as shown by the demonstrations.

To simplify the learning algorithm, we could preset the value of smoothing parameters σ^i. Essentially, as we will show later on, by presetting the value of σ^i, we transform a non-convex optimization problem into a quadratic optimization problem, which is convex and has a unique global minimum. Additionally, it makes the optimization computationally fast and tractable especially in higher dimensional problems. There is a trade-off in selecting values of σ^i. A too large or too small value for σ^i are not desirable as it makes the potential function over or under smoothed, respectively.

As a rule of thumb, we choose σ^i so as at least 5% of the data points remains within the 1-σ^i distance from each center $\boldsymbol{\xi}^i$.

There is a redundancy inherent in selecting the values of ϕ_0^i and $\boldsymbol{D}^i$. For example, consider we want the robot to move with a constant speed in some region in the task. Given the current robot state $\boldsymbol{\xi}$ and $\dot{\boldsymbol{\xi}}$, this can be achieved by setting Eq. (11) to zero. Since $\boldsymbol{S}^i$ are given, the contribution of the attracting terms $\sum_i \boldsymbol{\tau}^i_{attract}$ is fixed. However, there are infinite ways to select the nominal and damping terms to make Eq. (11) to zero. We could exploit this redundancy to better customize our control policy.

Essentially, we want our control policy $\boldsymbol{\tau}_c$ to apply a non-zero generalized force $\forall \boldsymbol{\xi} \in \mathbb{R}^d \setminus \boldsymbol{\xi}^*,\ \dot{\boldsymbol{\xi}} = \mathbf{0}$. In other words, we want the robot to be able to start the motion from static at any point except at the target, otherwise it may get stuck in a local minima. By inspecting Eq. (9), we could observe that at zero velocity, this force is solely given by the negative of the potential gradient, i.e. $-\nabla\Phi(\boldsymbol{\xi})$.

Since there is usually inaccuracy in modeling the physical properties of the robot, e.g. joints friction, it is even particularly important to impose a lower bound on the magnitude of the potential gradient. For example we notice that even on a high-performance robot such as KUKA LWR-IV, when controlling it in the operational space, a generalized force less than $2N$ barely moves the robots. Thus, when working on this robot, we need to at least have a lower bound of $2N$ on the potential gradient.

Putting together the above factors, we want to have a non-zero potential gradient across the demonstrations. The value of the potential gradient solely depends on the parameters ϕ_0^i. Hence, it allows us to split our learning algorithm into two phases: 1) learning ϕ_0^i to achieve the desired potential gradient at each training data point, and 2) learning $\boldsymbol{D}^i$ based on the estimated value of ϕ_0^i so as to generate the same velocity profile as the demonstrations.

4.1 Learning Parameters ϕ_0^i

Let us assume $\boldsymbol{\gamma}^i \in \mathbb{R}^d$ are the desired negative potential gradient at each point $\boldsymbol{\xi}^i$, i.e. $\boldsymbol{\gamma}^i = -\nabla\Phi(\boldsymbol{\xi}^i;\boldsymbol{\Theta})$. Note that depending on how $\boldsymbol{\xi}$ is defined (i.e. position, orientation, joint angle, or their combination), the unit of $\boldsymbol{\gamma}^i$ may be N, $N.m$, or a combination of them. The direction of each $\boldsymbol{\gamma}^i$ is given by $(\boldsymbol{\xi}^{i+1} - \boldsymbol{\xi}^i)/\|\boldsymbol{\xi}^{i+1} - \boldsymbol{\xi}^i\|$. Its magnitude is determined based on the characteristics of the robot and the desired safety standard, e.g. high enough to overcome the joints friction, and low enough not to cause any injury if the robot suddenly collides with a human. The value of $\boldsymbol{\gamma}^i$ directly determines the end-effector acceleration when starting from zero velocity. Without loss of generality, we consider constant magnitude for all $\boldsymbol{\gamma}^i$ for each experiment. This allows to have a more predictable robot motion throughout the task without limiting its performance. However, whenever it is necessary for the task, one can apt for variable magnitudes of $\boldsymbol{\gamma}^i$ with their value being selected based on the desired acceleration for the robot at zero velocity starting from different regions in the task space.

As the value of $\nabla\Phi(\boldsymbol{\xi})$ only depends on ϕ_0^i (see Eq. (12)), the optimization learning parameters $\boldsymbol{\Theta}$ is a vector created from concatenation of all ϕ_0^i, i.e. $\boldsymbol{\Theta} = [\phi_0^1 \ \cdots \ \phi_0^{\mathcal{T}}]$. An estimate of $\boldsymbol{\Theta}$ can be obtained by solving the following constrained quadratic optimization problem:[2]

$$\min_{\boldsymbol{\Theta}} J(\boldsymbol{\Theta}) = \frac{1}{\mathcal{T}} \sum_{i=1}^{\mathcal{T}} \|\nabla\Phi(\boldsymbol{\xi}^i;\boldsymbol{\Theta}) + \boldsymbol{\gamma}^i\|^2 \tag{13a}$$

subject to

$$\phi_0^{i+1} \leq \phi_0^i \qquad \forall i = 1..\mathcal{T},\ i \notin \Omega,\ i+1 \notin \Omega \tag{13b}$$

$$0 \leq \phi_0^i \qquad \forall i = 1..\mathcal{T},\ i \in \Omega \tag{13c}$$

$$\nabla\Phi(\boldsymbol{\xi}) = \mathbf{0} \qquad \boldsymbol{\xi} = \boldsymbol{\xi}^* \tag{13d}$$

where Ω is the set of indices that corresponds to the last point of each demonstration trajectory, which by construction is placed at the target point $\boldsymbol{\xi}^*$:

$$\Omega = \{i | \boldsymbol{\xi}^i = \boldsymbol{\xi}^*\} \tag{14}$$

The constraints given by Eqs. (13b) and (13c) are essential to ensure that ϕ_0^i are positive and energy consistent, i.e. for each demonstration trajectory the energy decreases by moving along the demonstration and always remains positive. Note that we use inequality instead of strict inequality to avoid unnecessary constraint to decrease ϕ_0^i if the two neighbor points $\boldsymbol{\xi}^i$ and $\boldsymbol{\xi}^{i+1}$ are identical. Equation (13d) also verifies that the gradient at the target point will be zero.

The optimization problem given by Eq. (13) has $\mathcal{T}$ parameters with $\mathcal{T}$ inequality constraints and d equality constraints. Although it may not be explicitly clear, Eq. (13) can be transformed into a constrained quadratic optimization problem after a few rearrangements (see Appendix A). Therefore, despite the high-dimensionality of this optimization problem, it can be efficiently solved within a few seconds using the state-of-the-art solvers such as cvxgen (Mattingley and Boyd, 2012).

Figure 7 shows an example of learning a 2D motion from a single demonstration trajectory composed of 500 data points. We choose $\sigma^i = 0.02$, $\boldsymbol{S}^i = 100N/m$, and $\|\gamma^i\| = 5N$. We start the optimization with an initial guess on the parameters ϕ_0^i by drawing uniformly from $[0\ \ 10]$. Obviously this is a bad initial guess and as can be seen in Fig. 7a the potential function includes several local minima. Right after one optimization iteration, all these local minima vanish and the potential attains its global minimum at the target (see Fig. 7b).

[2] Note that $\boldsymbol{\gamma}^i$ point at the negative direction of potential field gradient, i.e. $\nabla\Phi(\boldsymbol{\xi}^i;\boldsymbol{\Theta}) = -\boldsymbol{\gamma}^i$. This is why $\boldsymbol{\gamma}^i$ appears with positive sign in Eq. (13a).

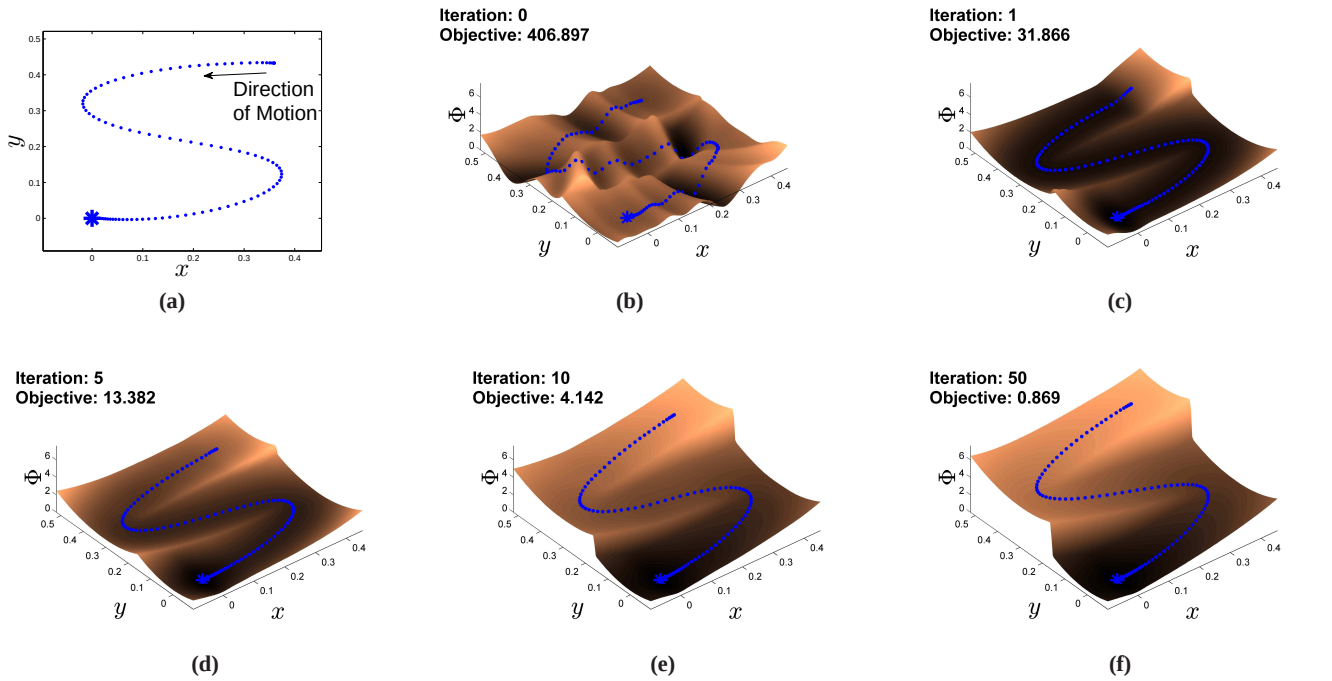


Fig. 7: An example of learning a 2D motion from a single demonstration trajectory composed of 500 data points (for clarity of the figure only every fifth points is shown). The initial guess on the parameters ϕ_0^i is obtained by drawing uniformly from $[0\ 10]$. The potential from the initial guess includes several local minima (see Fig. 7a). Right after one optimization iteration, all these local minima vanish and the potential attains its global minimum at the target (see Fig. 7b). The optimization terminates within 50 steps and achieve the desired potential gradient and stiffness.

The optimization iterates 50 steps to obtain the desired potential gradient and stiffness for this pattern. Due to the convexity of the optimization problem, the obtained solution is globally optimal.

Existence of a solution: A concern that might be risen is that whether there is any feasible solution to the optimization problem above. In other words, is it possible to determine a combination of ϕ_0^i that satisfies all the constraints? By integrating backward the traveled distance for each demonstration (i.e. from the target to the initial point), we could obtain a good initial guess for ϕ_0^i that satisfies Eqs. (13b) and (13c). However, ensuring existence of a solution that satisfies Eq. (13d) is non-trivial.

To address this concern, we introduce the notion of *virtual demonstration*. Let us define a set of indices $\mathcal{I}^v$, which includes all the centers $\boldsymbol{\xi}^i$ that are active at the origin (i.e. target point):

$$\mathcal{I}^v = \{i \in 1..\mathcal{T} \mid \tilde{\omega}^i(\mathbf{0}) > \epsilon\} \tag{15}$$

where $0 < \epsilon \ll 1$ is a very small threshold number. The set $\mathcal{I}^v$ contains $\mathcal{T}^v$ elements. We use the notation $\mathcal{I}^v_j$, $j \in 1..\mathcal{T}^v$ to refer to the j-th element of the set. We now define a set of virtual demonstration and append them to the original demonstration as follows:

$$\forall j = 1..\mathcal{T}^v \tag{16}$$
$$\boldsymbol{\xi}^{\mathcal{T}+j} = -\boldsymbol{\xi}^{\mathcal{I}^v_j}, \quad \dot{\boldsymbol{\xi}}^{\mathcal{T}+j} = -\dot{\boldsymbol{\xi}}^{\mathcal{I}^v_j}, \quad \ddot{\boldsymbol{\xi}}^{\mathcal{T}+j} = -\ddot{\boldsymbol{\xi}}^{\mathcal{I}^v_j},$$
$$\sigma^{\mathcal{T}+j} = \sigma^{\mathcal{I}^v_j}, \quad \boldsymbol{S}^{\mathcal{T}+j} = \boldsymbol{S}^{\mathcal{I}^v_j}, \quad \boldsymbol{D}^{\mathcal{T}+j} = \boldsymbol{D}^{\mathcal{I}^v_j},$$
$$\phi_0^{\mathcal{T}+j} = \phi_0^{\mathcal{I}^v_j}, \quad \boldsymbol{\gamma}^{\mathcal{T}+j} = \boldsymbol{\gamma}^{\mathcal{I}^v_j}$$

Equations (15) and (16) simply create a new data point for each center that is active at the origin. This results in having $\mathcal{T} + \mathcal{T}^v$ centers. For all $j = 1..\mathcal{T}^v$, the index of the new (virtual) data point is $\mathcal{T} + j$, which is virtually link to its counterpart with index $\mathcal{I}^v_j$ in the original demonstrations. The position, velocity, and acceleration of the virtual centers are negative of their corresponding counterparts in the actual demonstrations, while their remaining properties (stiffness, damping, etc.) are exactly similar.

With the addition of this new (virtual) data set to the original demonstrations, it can be shown that for each ϵ there is a small positive scalar $0 < \delta(\epsilon)$, where $\|\nabla\Phi(\boldsymbol{\xi}^*)\| \leq \delta(\epsilon)$. Note that $\delta(\epsilon)$ goes to zero as ϵ goes to zero. This relation holds for any demonstrations, ensuring that a feasible solution always exists within a $\delta(\epsilon)$ tolerance. By tuning the parameter ϵ we could define a tolerance on satisfying the optimization constraint given by Eq. (13d), which is a common requirement when doing numerical optimization.

Note that we train the control policy based on the original and virtual demonstrations. The use of virtual demon-

strations provides us with *theoretical proof* on availability of an optimal solution for any set of demonstrations. The addition of virtual demonstrations does not increase the number of optimization parameters because they are simply set to the values of their corresponding counterparts in the original demonstrations. Despite this, the user is not obliged to use virtual demonstrations as practically the optimization usually converges with only using the original demonstrations.

Figure 8 shows an example of learning a potential function from a 2D demonstration. For illustrative purpose, we consider a straight line motion composed of 100 data points. We set $\sigma^i = 0.025$ and $\epsilon = 10^{-5}$, which result in creating 12 virtual points. The original and virtual points are shown by red and blue circles, respectively. The desired potential gradients are defined as follows: $\boldsymbol{\gamma}^i = [5; 0]N,\ \forall i = 1..99$ and $\boldsymbol{\gamma}^{100} = [0; 0]N$. We consider the desired stiffness to follow a sine curve, i.e. $\boldsymbol{S}^i = s^i \boldsymbol{I}$, where $s^i = 200(\sin 2\pi \|\boldsymbol{\xi}^i\|)^2 + 100N/m$. The potential function is learned so as to reflect the desired behavior: its landscape becomes steep in regions with desired high stiffness and shallow in regions with low stiffness. The potential gradient at the center points is (approximately) constant with the norm equal to 5. This can be observed by looking at the distance between the energy levels. Note that in a small region around the target the potential gradient smoothly decreases to force the motion to stop at the target, which is necessary to ensure stability. The norm of the potential gradient at the target point is $\|\nabla\Phi(\boldsymbol{\xi}^*)\| = 2.47 \times 10^{-15}$N, indicating that the optimization constraint Eq. (13d) is satisfied with a very small tolerance. Henceforth, we use virtual demonstrations in all the experiments we report in this paper; however for clarity of graphs, we do not illustrate them.

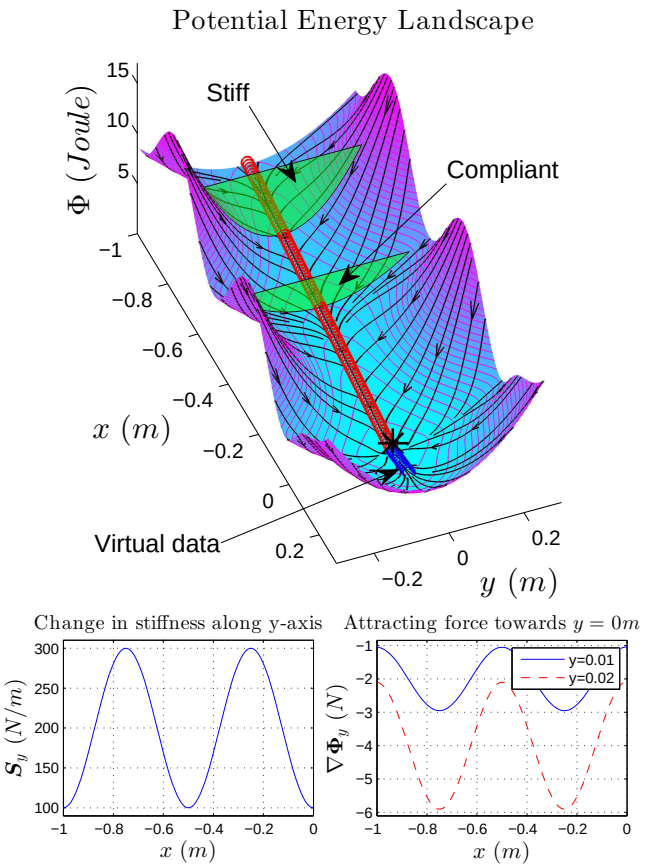


Fig. 8: Illustration of encoding variable stiffness with UMIC. For clarity of the figure, we consider a straight line nominal motion along the x-axis. The original and virtual points are shown by red and blue circles, respectively. In this example, the desired stiffness along the direction of attraction (i.e. y-axis) follows a sine curve (see bottom-left). As a result, the potential energy landscape becomes steep in regions with desired high stiffness and shallow in regions with low stiffness (see top). The figure on the bottom-right shows the attracting force towards the nominal motion at lines $y = 0.01$ and $y = 0.02$. As expected, this force increases proportionally to the stiffness and to the distance to the nominal motion.

4.2 Learning Damping Parameters

The velocity $\dot{\boldsymbol{\xi}}^i$ and generalized force $\boldsymbol{\tau}^i$ at each center point $\boldsymbol{\xi}^i$ are provided through the demonstrations. The value of the potential gradient at $\boldsymbol{\xi}^i$ can be directly computed based on ϕ_0^i as described in Section 4.1. Let us assume the damping matrices are in the form $\boldsymbol{D}^i = d^i \boldsymbol{I}$, then we could compute an optimal value for $\Xi = [d^1\ \cdots\ d^{\mathcal{T}}]$ by solving the following convex optimization problem:

$$\min_{\boldsymbol{\Xi}} J(\boldsymbol{\Xi}) = \frac{1}{\mathcal{T}} \sum_{i=1}^{\mathcal{T}} \|\tau^i + \nabla\Phi(\boldsymbol{\xi}^i) + \boldsymbol{\Psi}(\boldsymbol{\xi}^i, \dot{\boldsymbol{\xi}}^i; \boldsymbol{\Xi})\|^2 \quad (17a)$$

subject to

$$\varepsilon \leq d^i \qquad \forall i = 1..\mathcal{T} \quad (17b)$$

where $\varepsilon > 0$ is a lower bound on the value of the desired damping. Similarly to the previous section, this optimization problem is also quadratic and thus has a unique global solution, which can be obtained quickly. Figure 9 shows an example of learning ϕ_0^i and $\boldsymbol{D}^i$ for a J-shaped pattern from 7 demonstrations. As can be seen both the pattern and the velocity profiles are learned successfully.

Note that based on our experimental evaluation, the assumption $\boldsymbol{D}^i = d^i \boldsymbol{I}$ is not limiting and allows to perform a wide variety of tasks. However, in case it is necessary, the optimization problem given by Eq. (17) can be easily extended to the case where $\boldsymbol{D}^i$ is a diagonal matrix that has different eigenvalues along each dimension. The extension to have a full matrix $\boldsymbol{D}^i$ can also be achieved, though at the cost of solving a more complex optimization problem. A summary of the learning algorithm to train a UMIC model from demonstrations is provided in Algorithm 1.

5 Stability Analysis

The UMIC control policy given by Eq. (9) is by construction globally stable because it is modeled by a potential

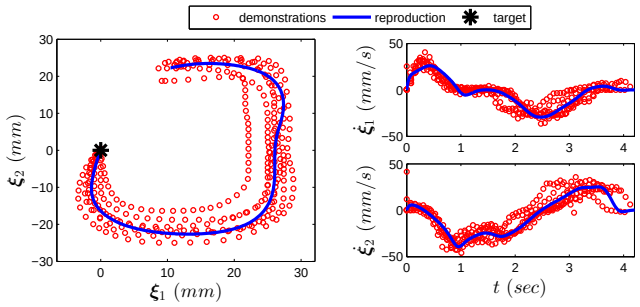


Fig. 9: An example of learning ϕ_0^i and $\boldsymbol{D}^i$ for a J-shaped pattern from 7 demonstrations. The thick blue line shows the reproduction of the task from an initial point on the top of the graph. As can be seen both the pattern and the velocity profiles are replicated successfully. Though not shown in the graph, this motion can also be generated starting from other points in the state space.

Algorithm 1 Learning UMIC from demonstrations

1: Collect a set of demonstrations as described in Section 2.1.
2: Set the parameters σ^i based on the task requirements.
3: Define desired magnitude of $\boldsymbol{\gamma}^i$ based on the task requirements.
4: Create virtual demonstrations as described in Section 4.1.
5: Estimate the parameters ϕ_0^i by solving the convex optimization problem given by Eq. (13).
6: Estimate the damping parameters $\boldsymbol{D}^i$ by solving the convex optimization problem given by Eq. (17).

function and a dissipative field (Khatib, 1986). It also implies a passive mapping from external force $\boldsymbol{\tau}_{ext}$ to $\dot{\boldsymbol{\xi}}$, hence ensuring that the robot remains stable in contact with any passive environment. The proof of stability is provided in Appendix B. This is a key feature as it *ensures global stability* (safety) of the task regardless of the situation of the robot and its surrounding environment. Note that by unifying motion generation and impedance control, the notion of tracking a reference trajectory is entirely removed from our approach. Hence, it is only sufficient to show the UMIC stability for the regulator case.

Furthermore, the learning algorithms proposed in Section 4 determine the parameters ϕ_0^i such that the target becomes the attractor of the potential function $\Phi(\boldsymbol{\xi})$. Thus, we could ensure local asymptotic stability of the system in a region close to the demonstrations. Although it is not possible to theoretically prove global asymptotic stability of the controller, as we will showcase through various experiments in Section 6, our controller is practically globally asymptotically stable (note the difference between stability and asymptotic stability). This is due to the fact that the terms $\boldsymbol{S}^i(\boldsymbol{\xi}-\boldsymbol{\xi}^i)$ attracts a point $\boldsymbol{\xi} \in \mathbb{R}^d$ that is far from demonstrations towards the nominal motion (i.e. the centers $\boldsymbol{\xi}^i$). Once $\boldsymbol{\xi}$ is sufficiently close to the demonstrations (i.e. within local asymptotic stability region), then it will converge to the target; hence the controller becomes globally asymptotically stable. Although we have not practically faced a spurious attractor in our experiments, if it happened, the user could overcome the problem by providing an additional demonstration that starts from the spurious attractor. In fact, this fits well within the context of active learning, where the robot could ask the user for help (providing more information) when it faces any issue during execution of a task.

6 Experimental Evaluations

We evaluate the performance of the proposed control policy via three sets of experiments. The first experiment is performed on the LASA human handwriting library (Khansari-Zadeh, 2011). This dataset contains thirty handwriting motions recorded from Tablet-PC, and has been used as a benchmark dataset to evaluate the performance of different movement generator algorithms (Khansari-Zadeh et al., 2013). With this experiment we illustrate the capability of our method in modeling complex motions. The second experiment includes a reach and place task on the 7-DoF LWR robot. Through this experiment we demonstrate how our controller allows to safely execute a task while being able to precisely place an object on the desired location. In addition, we showcase the suitability of our method for realtime execution on robots. The third experiments reports on performance quantification of the UMIC controller in encoding different stiffness profiles while following a same nominal motion. With this experiment we highlight that despite unifying motion generation and variable impedance control, our controller still allows the user to modify each part separately. This experiment is also performed on the LWR robot.

Note that in all our experiments the origin of the reference coordinates system is attached to the target.

6.1 Experiments on Human Handwriting Motions

The recorded motions in the LASA human handwriting library are two dimensional, i.e. $\boldsymbol{\xi} = [x; y] \in \mathbb{R}^2$. For each pattern, 7 demonstrations are provided by starting the motion from different initial positions (but fairly close to each other) and ending at the same final point. All the demonstrations have the same number of datapoints, and may intersect each other. Among the 30 handwriting motions in the library, 26 of them include one single pattern, three with two patterns, and one with three patterns, see Fig. 10. We start by averaging all the demonstrations for each pattern, hence obtaining a single demonstration for each one. Discussion on the advantage of performing this step is provided in Section 7.4.

In this experiment we manually set stiffness to linearly increase from $\boldsymbol{S}^0 = 100\boldsymbol{I}$ at the initial point of each demonstration to $300\boldsymbol{I}$ at the target point, where $\boldsymbol{I} \in \mathbb{R}^{2\times 2}$ denotes the identity matrix. We choose $\sigma^i = 5$, and $\|\boldsymbol{\gamma}^i\| = 500$ for all the demonstrations.

The collected demonstrations are shown with red thick line in Fig. 10. The gradient of the potential energy of the

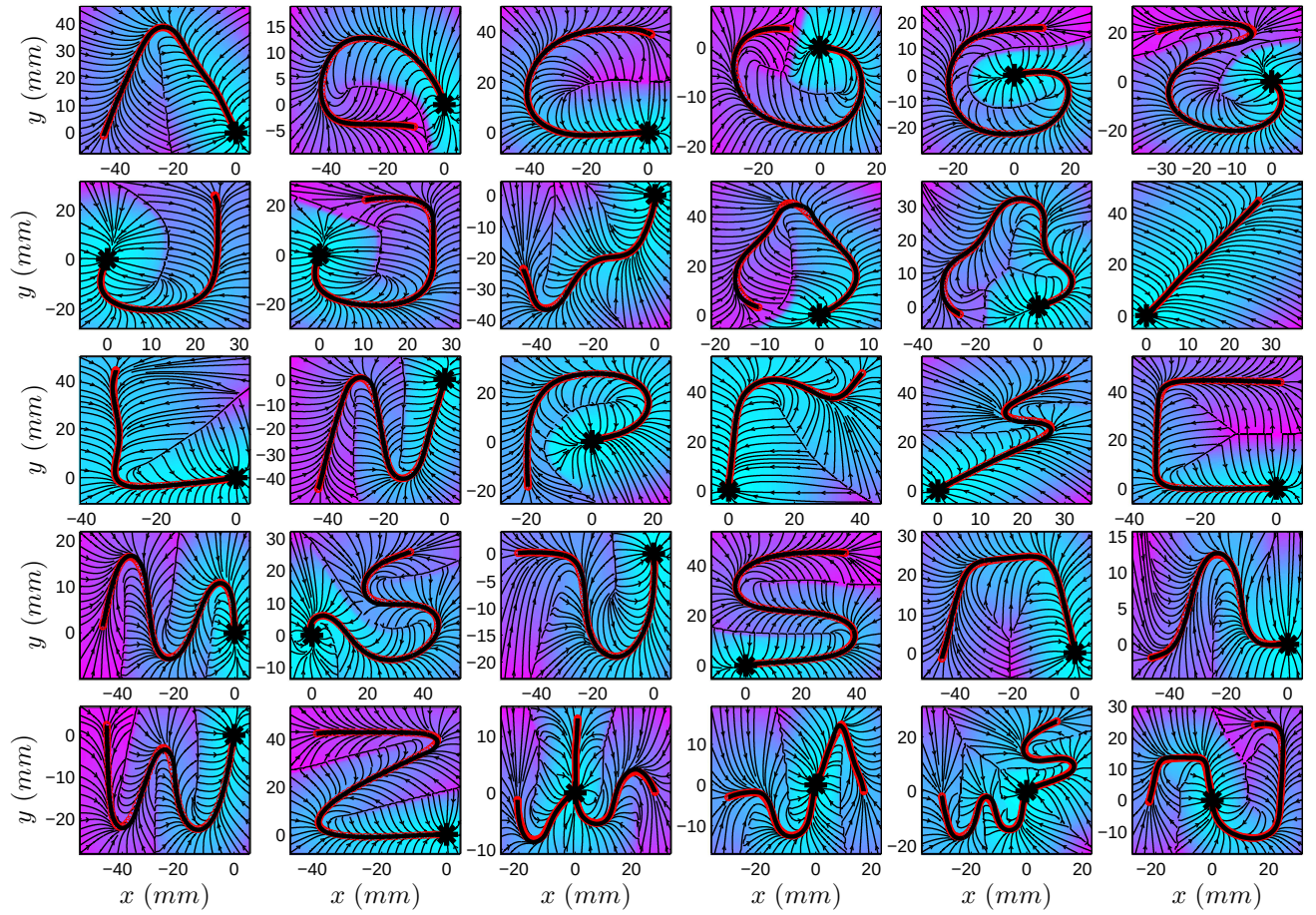


Fig. 10: Learning 30 human handwriting patterns using the proposed controller. In this graph, the potential energy is illustrated by color gradient. The white color indicates zero energy and it gradually increases towards blue. The target point is shown by a star. The thin solid lines indicates the potential energy gradient. Demonstrations are illustrated by a thick solid red line. The thick black line shows the motion when it is started at the initial point of the demonstration(s). We could see that the learned model precisely captures the motion from the demonstration(s) because the black line follows the red line throughout motion. For more information refer to Section 6.1.

trained models are superimposed on the demonstration. The thick black line shows the motion if it gets started from the initial point of the demonstration. As can be seen, the motions generated from the learned model matches the demonstrations, indicating that the nominal motion is being successfully captured by the trained model. The thin lines show the trajectories generated from the learned model starting from other initial conditions. We could see that all the models are asymptotically stable at the target. Furthermore, the main pattern is properly captured in each model and can be reproduced when the motion starts close to initial points of demonstrations. For multi-patterns models, UMIC is able to successfully capture the desired patterns as long as they are spatially separable (see the last four models in Fig. 10).

Note that our controller is time-invariant, and thus it derives the motion only based on the spatial information. This time-invariancy makes our controller inherently robust to temporal perturbations, hence increasing the controller's safety. In addition it frees us from all the problems associated with time-warping of trajectories to properly align them before the training. However, due to its time-invariancy, its generalization is also based on spatial situation. Thus, if we start the motion somewhere close to the middle of the demonstration (e.g. point $[-20; 40]$ in the first model), only half of the pattern will be reproduced. This kind of generalization could be a feature or a drawback depending on the application. If it is important to exactly reproduce the desired pattern starting from any point in the space, then it is necessary to place the origin of the frame of reference at the initial point (instead of the target, the way we consider it throughout this paper).

6.2 Experiment on the KUKA LWR Robot

In this experiment the robot is required to place a hammer into a toolbox in front of it. To better illustrate the safety feature of our controller, we significantly limit the robot's perception of the environment. In our experiment, the robot only knows the toolbox's position, and thus is not aware of any object that may come into its path. This experiment could be representative of situations where human and robot

Table 1: Functions used to determine $\boldsymbol{S}^i = s^i \boldsymbol{I}$ and σ^i in the position and orientation controllers. The variables $\boldsymbol{x}$ and $\boldsymbol{o}$ refer to the robot position and orientation, respectively.

	stiffness value s^i	**variance** σ^i
pos.	$100e^{-50\|\boldsymbol{x}^i\|_2^2} + 80e^{-20\|\boldsymbol{x}^i\|_2^2} + 20$	$\frac{1}{s^i} + 0.01$
orient.	$3e^{-100\|\boldsymbol{o}^i\|_2^2} + 2e^{-20\|\boldsymbol{o}^i\|_2^2} + 2$	$\frac{0.1}{s^i} + 0.01$

share the workspace. As estimation and prediction of human movements may not be possible in all situations, it is essential to have a safe controller such that even when the robot collides with the human, it does not harm her (similarly to the situation when two co-workers may occasionally bump into each other).

We use operational space control to model the task. We consider two controllers: $\boldsymbol{\tau}_c^p$ to control the end-effector position $\boldsymbol{x}$ and $\boldsymbol{\tau}_c^o$ to control its orientation $\boldsymbol{o}$. We use axis angle representation to model the orientation. For this task, we give a higher priority to the position controller and define $\boldsymbol{\tau}_c = \boldsymbol{\tau}_c^p + \mathcal{N}^p \boldsymbol{\tau}_c^o$, where $\mathcal{N}^p$ is the null-space matrix of the position controller. We start by first collecting some examples of the task. In total, we collect four demonstration trajectories. The collected trajectories are illustrated in Fig. 11a. Note that each demonstration captures information about both position and orientation. For clarity of the figure, we only show the orientation at the initial and final points.

We define the stiffness property to be low in most parts of the demonstrations to have a safe controller. We only increase it in a small region close to the target to improve the accuracy in reaching the target. To achieve this goal, we define the stiffness values using the exponential family functions (as stated before, one could also collect this information through human demonstrations). We choose the parameters σ^i to be inversely proportional to the desired stiffness at that point. Thus, points that have high stiffness have a smaller region of influence, hence they could not generate high force values (which could violate human safety). Table 1 summarizes the functions used to define stiffness and smoothing parameter for the position and orientation controllers. Based on the characteristics of the robot, we consider $\|\boldsymbol{\gamma}^i\| = 5N$ and $\|\boldsymbol{\gamma}^i\| = 1.5N.m$ for the position and orientation controllers, respectively. These values are high enough to overcome the joints friction, and low enough not to cause injury if the robot suddenly collides with a human or other objects in the environment.

Generalization to unseen situations: We use the trained models to execute the task starting from different initial conditions. The robot is controlled at 500Hz, which indicates that our controller is computationally fast (the whole computation is done in order of $10\mu s$ on an Intel Core i7 8×3.5GHz processor with 16GB of RAM). Figure 11b shows the robot motion starting from an arbitrary configuration using the learned model. The stiffness property (for the position controller) is also shown by superimposed belted-ellipsoids[3]. Note we show the stiffness only along the direction of attraction. It indicates how much the robot should resist if perturbed along the orthogonal directions to its motion. The robot interaction property along the direction of motion is better described by the gradient of the potential, which are indicated by green arrows. As expected the gradient remains constant in most parts of the motion and vanishes at the target.

Figures 11c and 11d show the execution of the task from 20 different initial conditions and their corresponding velocity profiles. As can be seen the robot can reach the target in all cases with precision in order of millimeters (mean: 5.4mm, standard deviation: 1.2mm). This precision is more than necessary for the task at hand; however, if required, better accuracy can be achieved by increasing $\boldsymbol{S}^i$ at points close to the target. The way the trajectories converge to each other right before the target is also noteworthy, which is due to the increase in the local stiffness (see Fig. 11b for visualisation of stiffness). If we increase the stiffness further away from the target, its effect will also be reflected on the trajectories by converging to each other earlier.

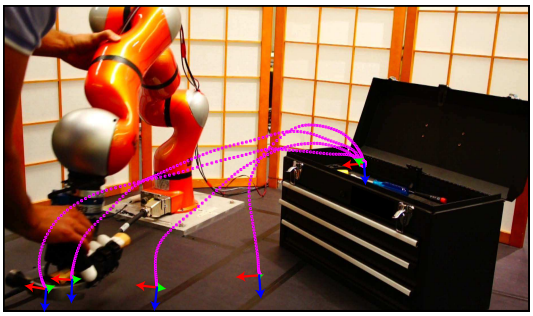
Safety in human-robot interaction: Figure 12 illustrates the safety feature of our controller, in which the robot collides with the user during the task execution. The robot is completely unaware of the user's presence. Furthermore, the controller does not rely on force/torque measurements during the task execution. We run 10 trials of this experiment starting the robot from different initial conditions and having the user (intentionally) collide with the robot in different courses of the task. The impact force in all the trials are low: $f_{max} = 4.88 \pm 0.73$N.

Note that in this experiment we considered a worst-case scenario and showed that the robot is still safe to operate despite inevitable collisions with the human. In fact the proposed controller brings robots one step closer towards sharing workspace with humans. Nevertheless, whenever there is additional sensory information about the objects in the environment, our controller can be equipped with the state-of-the-art techniques to avoid collision. We encourage the reader to check the accompanying video for an illustration of the result, the safety feature of our controller, and more experiments (which for brevity were not reported here).

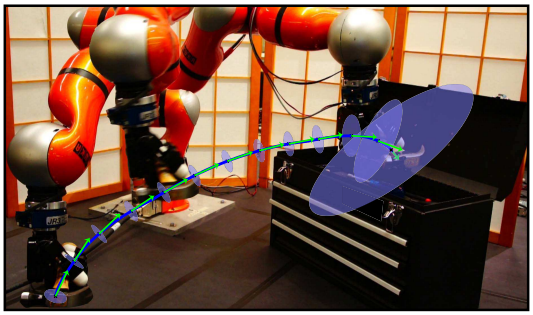
6.3 Evaluation of Stiffness and Motion Control

In Section 3 we stated that the UMIC control policy allows to separately modify the impedance properties with no or little change in the nominal motion. In this section we quantify this property via a robot experiment on the LWR robot.

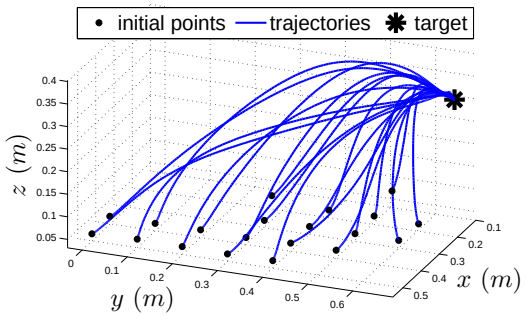
[3] As we consider diagonal stiffness matrices of the form $\boldsymbol{S} = s^i \boldsymbol{I}$, belted -ellipsoids are in fact circles.



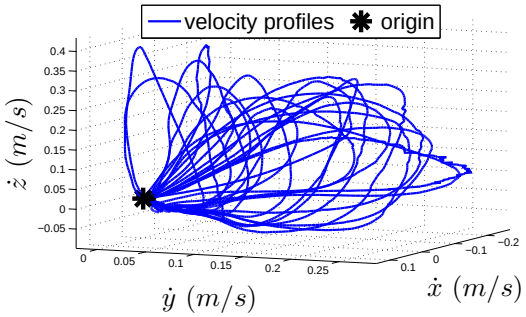
(a) Collection of four demonstrations through kinesthetic teaching.



(b) Execution of the task from an arbitrary initial configuration.



(c) Generalization of the task to unseen situations.



(d) Velocity profile of motions for the motions in (c).

Fig. 11: In this experiment, the LWR robot places a hammer onto a toolbox. In **(b)**, circles indicate the robot stiffness on the perpendicular direction to the motion, and green vectors show the gradient of the potential energy. As can be seen in **(c)**, the robot could reach the box from various initial configurations (accuracy in order of millimiters). Please refer to Section 6.2 for more information.

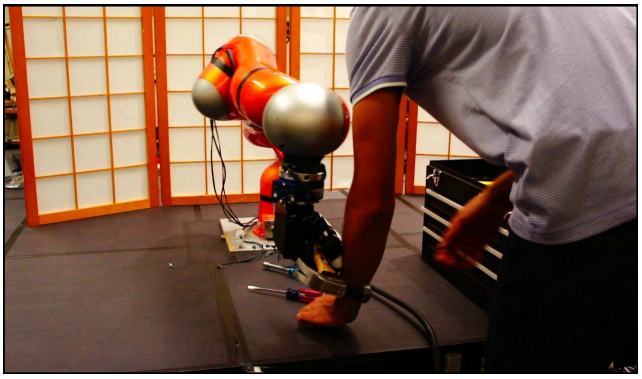


Fig. 12: In this experiment, the robot is unaware of any object in the workspace except the toolbox. It collides with the user during the execution of the task; however, due to its compliancy, the max impact force in all the trials is low $f_{max} = 4.88 \pm 0.73$N.

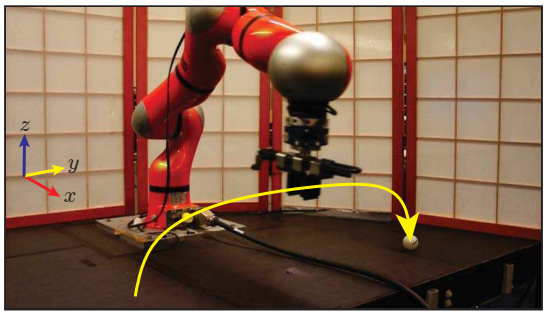
In this experiment the robot is taught to perform a reaching motion as depicted in Fig. 13a. We train this motion with 6 different stiffness profiles that are shown in Fig. 13b. We also set $\|\boldsymbol{\gamma}(\boldsymbol{\xi})\| = 10N$ and $\sigma^i = 0.01$ for all the centers (note that $\boldsymbol{\xi} = [x\ \ y\ \ z]^T$).

Note that as discussed in Section 6.2, in reaching tasks it is considered optimal to increase the stiffness toward the end of movement to cope with model uncertainty and thus to increase accuracy in reaching the target point. In this experiment, we consider a bell-shaped profile for stiffness for two reasons: 1) to illustrate that even at low stiffness our controller can still be accurate (see Fig. 13c), and 2) to allow us to evaluate the effect of stiffness on the controller performance through perturbing the robot in the middle of the motion (we will elaborate this experiment later on in this section).

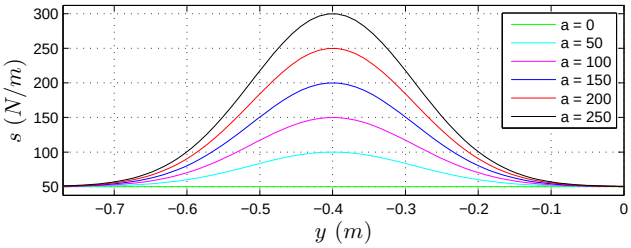
We execute the motion with the same initial ($\pm 1mm$ error) and target points for each of the trained models. Figure 13c shows the robot trajectory and velocity profiles in these 6 runs. We could observe that the nominal motion between these 6 models matches well, though each of them encodes a different stiffness profile.

We further compare these 6 trained models in collision with an unperceived object (see Fig. 14a). We rigidly fixed the object at $y = -0.4m$. We quantify two types of impact force: 1) tha maximum impact force $\boldsymbol{f}_{max}$, and 2) the steady-state impact force $\boldsymbol{f}_{ss}$. Generally $\boldsymbol{f}_{max} > \boldsymbol{f}_{ss}$ due to the momentum of the robot at the time of impact. The importance of evaluating $\boldsymbol{f}_{max}$ is that it includes the accumulated effect of the robot dynamics from the initial to the impact time. We run each trained model 10 times with the same initial ($\pm 1mm$ error) and target points, obtaining 60 trials in total.

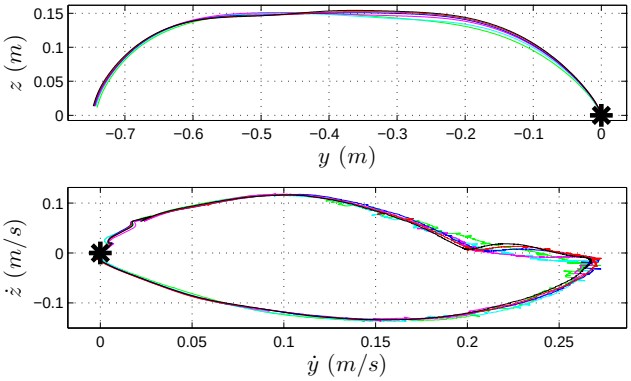
The statistics of the impact forces across all these trials is shown in Fig. 14c. Despite great changes in stiffness between the trained models, there is a small difference in the impact forces (i.e. $< \%5$). Note that in this experiment



(a) The robot is trained to perform a reaching motion.

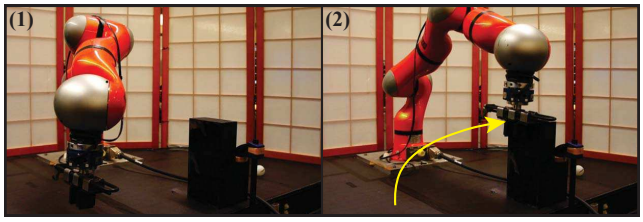


(b) We consider 6 different stiffness profiles along the demonstration. Each stiffness profile $\boldsymbol{S}(\boldsymbol{\xi}) = s(\boldsymbol{\xi})\boldsymbol{I}$ is defined by: $s(\boldsymbol{\xi}) = 50 + ae^{-40(y+0.4)^2}$, $a \in \{0, 50, 100, 150, 200, 250\}N/m$.

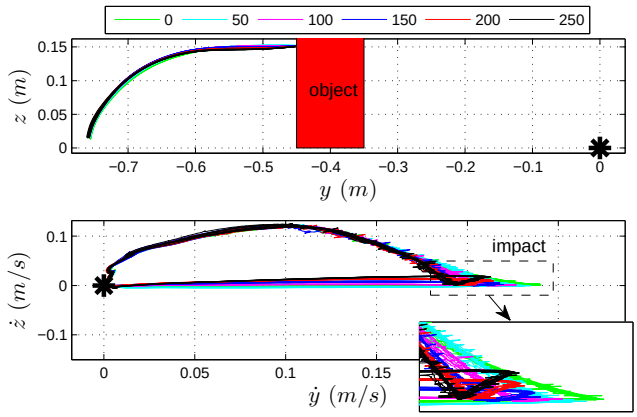


(c) Execution of the motion with the same initial ($\pm 1mm$ error) and target points six times, each of which with one of the trained models. For clarity, only the projected motion on the y-z plane illustrated as the motion along x-axis is negligible.

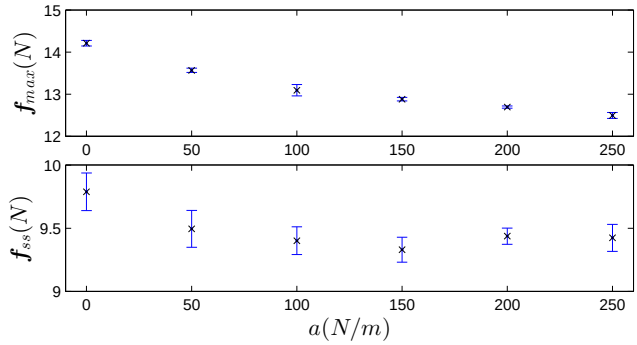
Fig. 13: Performance evaluation of UMIC in encoding six different stiffness profiles, while following the same nominal motion. Note that $\boldsymbol{\xi} = [x \;\; y \;\; z]^T$. Please refer to Section 6.3 for further information.



(a) Placing a rigidly fixed object on the robot path to evaluate the performance of the trained models at collision.



(b) Trajectory and velocity profile of the robot in the experiment with collision to an unperceived object. Each model is executed 10 times, running 60 trials in total.



(c) Maximum $\boldsymbol{f}_{max}$ and steady-state $\boldsymbol{f}_{ss}$ impact force for each trained model. The parameter a in the stiffness profile is defined in Fig. 13b.

Fig. 14: Performance evaluation of the 6 trained UMIC model in Fig. 13 at collision with an unperceived object.

the object is posed so as the collision occurs only along the direction of the motion. Therefore idealistically we should have $\|\boldsymbol{f}_{ss}\| = \|\boldsymbol{\tau}_{nominal}\| = \|\nabla\Phi(\boldsymbol{\xi}^c)\| = \|\boldsymbol{\gamma}\| = 10N$, where $\boldsymbol{\xi}^c$ refers to the contact point. However in this experiment the steady-state contact force is less than the trained value $10N$ due to the fact that a small part of the applied force is used to overcome the (unmodelled) joints friction of the robot. In addition, as shown in Eq. (12), $\|\boldsymbol{\tau}_{nominal}\|$ depends on both ϕ_0^i and stiffness values $\boldsymbol{S}^i$. Therefore, a change in desired stiffness would affect the optimization problem given by Eq. (13) and thus yields a different solution. This is why in Fig. 14c we observe different values for $\|\boldsymbol{f}_{ss}\|$ for each model.

The results on $\boldsymbol{f}_{max}$ may also seem counterintuitive because one may expect $\boldsymbol{f}_{max}$ should increases as stiffness increases. First note that idealistically $\boldsymbol{f}_{max}$ should remain fixed across all the models. However, the difference in values of $\boldsymbol{f}_{ss}$ also directly affects $\boldsymbol{f}_{max}$. Furthermore, $\boldsymbol{f}_{max}$ also depends on the velocity profile during the motion. In models with low stiffness, the robot slightly deviates from the nominal motion due to uncertainties in the robot model. Therefore, in these models both $\boldsymbol{\tau}_{nominal}$ and $\boldsymbol{\tau}_{attract}$ are affecting the motion and thus the robot achieves slightly higher velocity. By inspecting Fig. 14b, we could observe that the impact velocity is slightly higher for models with lower stiffness, describing why the the max impact force is

higher in these models. As stiffness increases, these deviations are more suppressed, and thereby the robot more precisely follows the nominal motion. This can also be verified by noticing that the slope of the $\boldsymbol{f}_{max}$ plot converges to zero as the stiffness increases.

7 Discussion

7.1 UMIC vs. other Varying Impedance Control Schemes

UMIC provides us with a time-invariant variable impedance controller to control the robot. There are however two other types of variable impedance controls that are commonly used in literature for tracking a *reference trajectory*: 1) *Time-varying* Variable Impedance Control (TVIC) (Buchli et al., 2011; Ferraguti et al., 2013; Kronander and Billard, 2013), and 2) *State-varying* Variable Impedance Control (SVIC) (Ott, 2008). There are a number of structural differences between UMIC, TVIC, and SVIC, which make each of them more suitable in certain applications. In this section, we aims at providing an objective comparison between these three approaches.

TVIC and SVIC can be used in two modes, regulator and tracking modes. The latter is a more generic form, where a time-indexed trajectory $\boldsymbol{\xi}^d(t)$ can be tracked. In our comparison we consider this model as it is closer to the UMIC[4]. When tracking a time-indexed trajectory, TVIC and SVIC control laws are as follows:

$$\boldsymbol{\tau}_{\text{TVIC}} = \boldsymbol{M}(\boldsymbol{\xi})\ddot{\boldsymbol{\xi}}^d(t) - \boldsymbol{D}(t)(\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}^d(t)) - \boldsymbol{S}(t)(\boldsymbol{\xi} - \boldsymbol{\xi}^d(t)) \tag{18}$$

$$\boldsymbol{\tau}_{\text{SVIC}} = \boldsymbol{M}(\boldsymbol{\xi})\ddot{\boldsymbol{\xi}}^d(t) - \boldsymbol{D}(\boldsymbol{\xi})(\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}^d(t)) - \boldsymbol{S}(\boldsymbol{\xi})(\boldsymbol{\xi} - \boldsymbol{\xi}^d(t)) \tag{19}$$

Note that, as mentioned before, we consider an impedance control law that does not reshape the inherent inertia of the robot. However, the results herein extend also to the case where a desired apparent inertia is implemented via force-feedback. To better illustrate the difference between these approaches, we consider a toy example, where a point-mass robot is required to move on a straight line to reach a goal position. The robot mass is $1kg$. Without loss of generality, let us assume the initial and final points are at $\boldsymbol{\xi}^0 = [-1; 0]$ and $\boldsymbol{\xi}^* = [0; 0]$. We designed a smooth trajectory $\boldsymbol{\xi}^d(t)$ composed of 501 points between these two points. This trajectory and its velocity profile are illustrated in Fig. 15a. We use this trajectory as the reference trajectory in TVIC and SVIC, and as the training demonstration in UMIC. We also consider the following stiffness and damping profiles in the domain $-1 \leq \boldsymbol{\xi}_1 \leq 0$, where $\boldsymbol{\xi}_1$ refers to the first component of the vector $\boldsymbol{\xi}$: $\boldsymbol{S}(\boldsymbol{\xi}) = s(\boldsymbol{\xi})\boldsymbol{I}$, $s(\boldsymbol{\xi}) = -400\boldsymbol{\xi}_1^2 + 500N/m$, and $\boldsymbol{D}(\boldsymbol{\xi}) = 2\sqrt{s(\boldsymbol{\xi})}\boldsymbol{I}\ N.s/m$. Figure 15b-left illustrates the $s(\boldsymbol{\xi})$ in the range between $-1 \leq \boldsymbol{\xi}_1 \leq 0$. Since TVIC defines the impedance properties as a function of time, we compute the equivalent of $s(\boldsymbol{\xi})$ in the time domain based on the desired trajectory $\boldsymbol{\xi}^d(t)$ (see Fig. 15b-right). Unless otherwise specified, throughout this section we use the term 'desired behavior' to refer to 'desired tracking performance' for TVIC and SVIC, and to 'following the nominal motion' for UMIC.

Safety to presence of unperceived objects: We first evaluate the difference between TVIC, SVIC, and UMIC in the presence of an unperceived obstacle. When robots work in unstructured environment, tracking all the objects in the work space is non-trivial. Moreover, installation of tactile and force sensors throughout the robot body is currently costly and difficult. Hence, it is essential to have a control law that is robust and safe when colliding with an unperceived object.

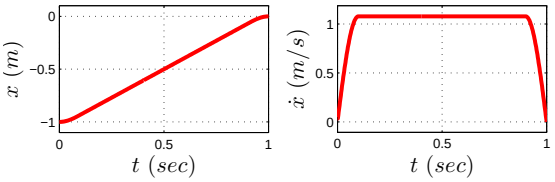
Figure 15c shows a scenario where an object is placed at $x = -0.5m$. Since both TVIC and SVIC relies on tracking a time-indexed trajectory, they start applying a continuously increasing force after the collision with the obstacle. This force reaches a peak that is more than 10 times larger than what the robot experience in the absence of obstacle. This force could potentially result in the robot breakage or damaging the collided object.

Note that since in SVIC the stiffness property is state-varying, the increase in force linearly increases with the $\|x^d - x\|$, whereas in TVIC the desired stiffness also increases with time (in this example). This is the reason why the applied force in TVIC is even larger than SVIC. In contrast, UMIC is a time-invariant system and thus when gets stuck behind the obstacle, it applies a constant low force based on the designed parameters. In this example, we use $\boldsymbol{\gamma} = 20N$, and thus the applied force is $20N$.

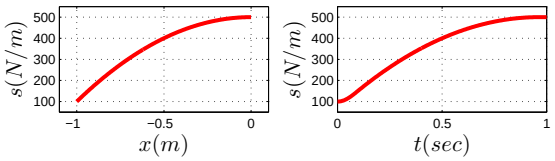
TVIC and SVIC-based methods could adopt a heuristic to avoid inducing a large force. An example of such heuristics is to threshold the control command at a desired (low) value. Although this heuristic could alleviate the problem of large contact force, it may not be suitable for tasks that require continuous contact with the environment. To perform these tasks properly, one should apt for a variable thresholding strategy. An example of such heuristic is to change the value of the threshold based on the end-effector position. Such heuristics could be hand-tuned to work in practice; however, providing a theoretical proof for such cases is difficult (if not impossible).

Another behavior that can be observed in time-varying approaches is that there is an instant jump in the robot motion right after the obstacle clears the path. This jump is due to the fact that the robot tries to catch-up with the reference trajectory, and as a result it induces a large acceleration and skips a portion of the trajectory. In contrast, when using a time-invariant approach, the robot follows the desired path

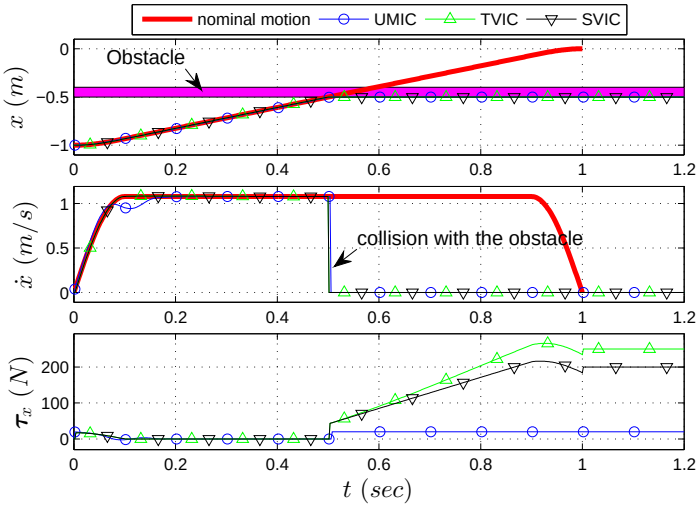
[4] Regulator case simply results in a motion that directly goes to the target point. Thus, it does not have the capability to follow complex patterns such as the ones illustrated in Fig. 10.



(a) The desired trajectory and its velocity profile along the x-axis used for tracking in TVIC and SVIC. Note that we define $\boldsymbol{\xi}^d = [x^d; y^d]$ and $y^d = \dot{y}^d = 0$. This trajectory is also used as the training data for UMIC.



(b) The desired state-varying stiffness (left) and its equivalent time-varying stiffness computed based on the desired trajectory.



(c) Behavior of the system in the presence of an unperceived obstacle.

Fig. 15: Comparison between TVIC, SVIC, and UMIC in the presence of an unperceived obstacle. Since both TVIC and SVIC relies on tracking a time-indexed trajectory, they start applying a continuously increasing force after the collision with the obstacle. In this example, this force reaches a peak that is more than 10 times larger than what the robot usually exerts in the absence of obstacle. In contrast, UMIC is a time-invariant system and thus when gets stuck behind the obstacle, it applies a constant low force based on the designed parameters. In this example, we use $\boldsymbol{\gamma} = 20N$, and thus the applied force is $20N$.

when the obstacle is cleared without imposing any large acceleration. We would like to point out that depending on the task, the aforementioned behavior can be seen as a feature or shortcoming and thus it is up to the user to choose the proper strategy based on the requirement of the task.

Motion Complexity: Both SVIC and TVIC are capable of tracking a complex-shaped pattern as long as its stability can be ensured. Since UMIC tries to build a generic model of the task as a time-invariant system, it imposes some constraints on the type of motion that can be modeled: the motion should be clearly definable in the state-space. More specifically, UMIC cannot model any pattern that has self-intersection due to the ambiguity that arises at the intersection point. Similarly, it cannot model cyclic patterns or those that have overlapping segments. One heuristic to model these types of motions is to decompose them into two or more segments and model each part with a separate UMIC model. However, special consideration should be taken at switching between the segments. Note that since UMIC is based on second order differential equations, the continuity in velocity can be preserved at the switching points.

We would like to point out that UMIC is a primitive-based approach and should not be confused with path planning techniques. Primitives should be considered as building blocks from which more advanced robot tasks can be formed (Schaal, 1999). For example, consider the task of lawn mowing. Instead of modelling the lawn mower pattern as a single primitive, this pattern should be decomposed into a set of meaningful segments (e.g. straight line motion, 90 degrees turn, etc.), each of which can be modeled as a UMIC primitive.

Response to perturbations: Figure 16 shows the behavior of all the three control policies when perturbed along the y-axis. The perturbation imposes the velocity of $1m/s$ along the negative y-axis at $t = 0.2s$. All the 3 controllers successfully recover from this perturbation and return to the desired behavior. Thus, in this regard they all behave similarly if the perturbation happens along the direction perpendicular to the desired direction of motion. Note that the lower value of $\boldsymbol{\tau}_y$ in UMIC is due to its lower damping value compared to TVIC and SVIC (recall that the damping values are learned in UMIC).

Stability analysis: Stability of both TVIC and SVIC need to be evaluated based on the given desired trajectory. In general this evaluation is hard and requires thorough consideration. Furthermore, it is a time-consuming process as one needs to redo the evaluation if the desired trajectory needs to be significantly modified. Hence, this imposes a major limitation on the applicability of TVIC and SVIC in dynamic tasks, where objects in the workspace may move and thus the robot needs to adapt accordingly. UMIC is by construction designed to be globally stable (i.e. bounded-input-bounded-output) as long as the continuity assumption is not violated. In all our experiments, with a control frequency equal or greater than 500Hz, this assumption remained valid. This is an important feature as it ensures the task safety in the robot workspace instead of around a reference trajectory. To obtain asymptotic stability at the target, we need to learn the

UMIC parameters by solving two constrained convex optimization problems. Compared to TVIC and SVIC, this is an additional step that needs to be taken.

Adaptation to changing environments: The ability to adapt to changes in the environment is an important feature for robots that are going to operate in uncontrolled workspace. In this regard, following a time-indexed trajectory imposes a great limitation on the robot, as this reference trajectory may not be adapted in realtime. In addition, even if the robot is equipped with a fast planning technique that could create a new reference-trajectory on-the-fly, there are still some concerns about stability of the variable impedance controller around this reference trajectory (which is hard to verify). By integrating motion generation and variable impedance control, UMIC can adapt instantly to changes in the environment by taking a new path to the target.

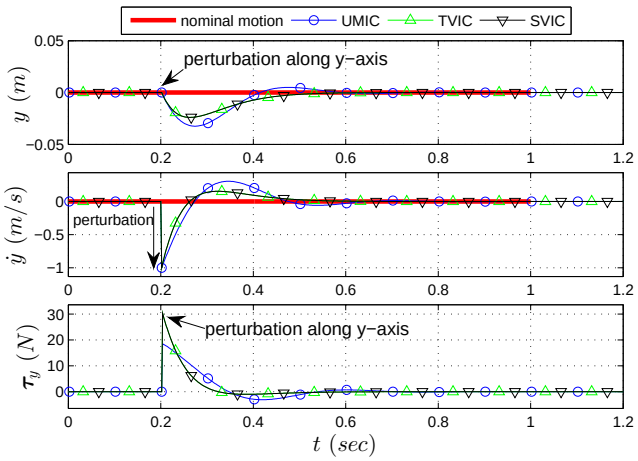


Fig. 16: Comparison between TVIC, SVIC, and UMIC when perturbed along the direction perpendicular to the desired direction of motion. In this example the desired direction of motion is along x-axis and a velocity perturbation of 1m/s is applied along negative y-axis. All the 3 controllers successfully recover from this perturbation and return to the desired behavior.

7.2 UMIC vs. Trajectory Tracking

The UMIC controller should not be confused with a tracking controller. Considering our toy example in Fig. 15c, if the motion starts at $x = -0.6m$, a tracking controller first pushes the robot towards the beginning of the reference trajectory at $x = -1m$, trying to reproduce the whole desired motion. In contrast, the UMIC controller just pushes the robot towards the target point $x = 0$ following the nominal motion, ignoring a part of motion from $x = -1$ to $x = -0.6m$. If it is important to exactly reproduce the desired pattern, then we could place the origin of the frame of reference at the initial point (instead of placing it on the target).

Furthermore, although a time-indexed trajectory can be used as a reference trajectory to train UMIC, the centers in UMIC should not be misunderstood with points in a time-indexed trajectory. A point in a reference trajectory only appears at a certain time in a time-space coordinates system, and disappears after that, whereas the centers in UMIC are always present in a time-space coordinates though their effect might be negligible due to the distance to the query point.

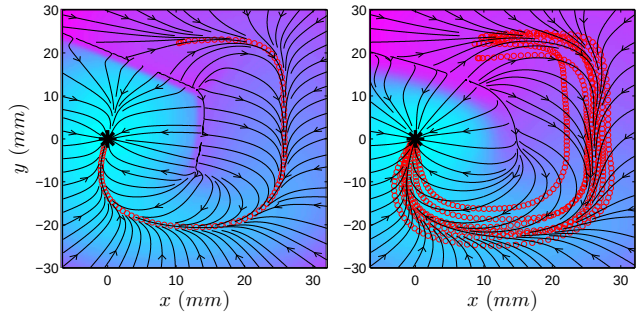
Moreover, the centers $\boldsymbol{\xi}^i$ in the UMIC are not via-points. By definition, a via-point imposes constraints on the robot to pass through it. The time order of via points is also important and they cannot be skipped. Therefore, a via-point approach can be pictured as a set of overlapping funnels, in which the motion starts from one funnel and is guided to the next funnel at the via-point (Khansari-Zadeh, 2012). Hence, it is represented in the time-space coordinates system. In UMIC, the motion is not forced to pass through the centers even when high-stiffness is used. There is no explicit time order for them and thus some of them may be skipped depending on the state of the robot. Considering Fig. 9, one can clearly distinguish the difference between UMIC and a via-point approach, especially in the presence of multiple demonstrations.

7.3 Choice of Reference Frame

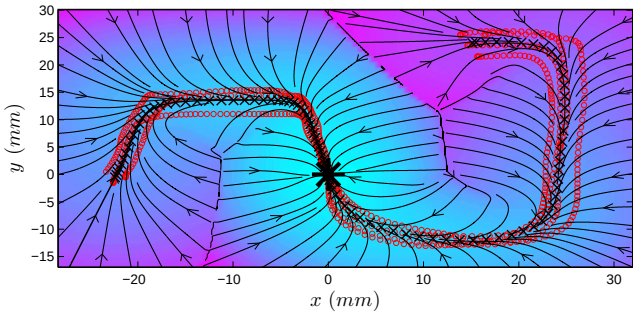
In all our experiments the origin of the reference coordinates system is attached to the target, i.e. $\boldsymbol{\xi}^* = \mathbf{0}$. All the center points $\boldsymbol{\xi}^i$ as well as the robot motion are defined with respect to this frame. Such representation makes the parameters of the control policy invariant to changes in the target position (Khansari-Zadeh and Billard, 2011). Furthermore, it simplifies the notation and reduces the number of operations when computing the torque command without imposing any limitation.

7.4 Multiple Demonstrations

Let us consider an example where the robot should follow a J-shaped pattern (see Fig. 17a). The user could provide one or several demonstrations of this pattern; both can be used to train the UMIC control policy. When providing several demonstrations of the same pattern, the learning algorithm finds a compromise across all the demonstrations, hence can better handle the inevitable noise in each demonstration. However, in this case none of the demonstrations is representing the nominal motion, which could make it difficult for the user to debug. Alternatively, one can generate a single main demonstration out of the provided demonstrations, and solely use that to train the UMIC controller. By doing so, we could take the advantage of using several demonstrations to handle the noise, while having a simple model that can be easily debugged. This procedure can be used when there are several demonstrations of a single pattern that are sufficiently close to each other.



(a) One or several demonstrations can be provided to train a single pattern. **Right:** When there are several demonstrations, it is easier to handle the noise-inherent in each demonstration. However, it is more difficult to determine the nominal motion (in contrast to the single demonstration). **Left:** To have the best of both worlds, we could first generate a reference demonstration out of the provided several demonstrations, and use that to train the UMIC controller.



(b) When there are several demonstrations of multiple patterns, first we generate a reference demonstration for each pattern (shown with black cross), and then train the UMIC controller based on these reference demonstrations. In this example, the J and the trapezoid patterns are provided with four and three demonstrations, respectively.

Fig. 17: Learning single and multiple patterns from demonstrations. In this example, the target point is shown by a black star, and demonstrations are illustrated by red circles.

When there are more than one patterns, we could create a reference demonstration out of the corresponding demonstrations of each pattern, and then use these reference demonstrations to train the controller. For example, Fig. 17b shows a controller composed of two patterns, each of which is represented by a single reference demonstration. In this example, the J and the trapezoid patterns are provided with four and three demonstrations, respectively.

7.5 On the Topic of Safety

This paper approaches the topic of 'safety' through employing a variable impedance strategy. In essence, the robot compliance plays an important role in enhancing safety to the human user (Kishi et al., 2012; Wolf and Hirzinger, 2008; Zinn et al., 2004). Using low stiffness is especially helpful in regions where there is a large uncertainty in perceiving the environment and thus a high chance of collision with unperceived objects. Compliance is also an adequate countermeasure to protect the robot during rigid and fast impacts with hard surfaces (Wolf and Hirzinger, 2008). We would like to point out that solely using a low-stiffness controller does not guarantee human safety when the robot is operating at high speed. The peak impact force is a function of the impact velocity and is larger than the steady-state contact force (see for instance Fig. 14c). Therefore, special consideration should be taken if the robot is moving at high speeds, and thus one should adopt a collision detection strategy and appropriate reaction methods to ensure human safety (Haddadin et al., 2008).

8 Related Work

This paper lies at the intersection of four research directions, namely: potential fields, movement primitives, impedance control, and robot learning. In Sections 8.1 to 8.3 we provide an overall account of research in the first three topics. In Section 8.4 we review in detail the most relevant work to this paper, which in fact address the problem of learning potential fields, primitives, and/or (variable) impedance controllers.

8.1 Potential Fields

The Potential field approach is one of the earliest works on feedback motion planning (Khatib, 1986). In this approach, the robot is considered as a particle, and its workspace is described by a global potential function that its gradient guides the robot to the target point. More specifically, the global potential function is defined as a sum of an attractive potential function located at the target, and a set of repulsive potential functions representing obstacles. The direction of the movement (i.e. the gradient direction) is thus governed by the net force induced due to the presence of all these fields. Potential functions are subject to local minima, i.e. they cannot ensure the target is always reachable.

A potential function that is free from local minima is called navigation function (Rimon and Koditschek, 1992). Harmonic potential functions (Kim and Khosla, 1992) and Fast Marching Square (Gomez et al., 2012) are two known navigation functions. Harmonic Potential functions are inspired by the description of the dynamics of some physical processes such as heat transfer or fluid flow. Despite ensuring global convergence to the target, construction of an exact navigation function is limited to simple environments with obstacles of specific shapes. Approximate methods based on discretized space overcome this limitation but at the cost of being computationally more expensive (Brock et al., 2008). Potential fields is a well studied research topic, and review of all existing approaches is beyond the scope of this paper. In Section 8.4 we review the works that are related to this paper, i.e. those that address the problem of learning potential functions. We refer interested readers to (Khansari-Zadeh, 2012; Koditschek, 1989) for a more general overview of potential fields approaches.

8.2 Movement Primitives

To perform a wide variety of tasks in unstructured environments, the idea of movement primitives has been investigated in robotics in the past two decades (Billard et al., 2008; Schaal, 1999). Movement primitives in essence can be seen as a basis from which more advanced robot tasks can be formed by sequencing or superimposing them. Each movement primitive codes a behavior (such as reaching for a cup, swinging a golf club, etc.) with a set of autonomous or non-autonomous differential equations (Billard et al., 2008; Ijspeert et al., 2002; Khansari-Zadeh and Billard, 2011). These techniques are often referred to as Dynamical System-based approaches since they directly define a robot motion with a differential equation (as opposed to the potential field approaches that first define an energy function and then take its gradient to generate the motion).

Among dynamical system approaches Dynamic Movement Primitives (DMP) (Ijspeert et al., 2002; Muelling et al., 2013; Ude et al., 2010), Stable Estimator of Dynamical Systems (Khansari-Zadeh and Billard, 2011) and its variants (Khansari-Zadeh and Billard, 2012, 2014), and Gaussian Mixture Methods (Calinon et al., 2010a, 2011; Gribovskaya et al., 2010) are noteworthy. The movement primitives approaches operate in 'closed-loop' and thus provide realtime adaptation to changes in dynamic environments. The major body of work in movement primitives has been focused on free-space motions and thus do not consider robot interaction with the environment. More recently, a few works aim to introduce interaction control into movement primitives. These works will be reviewed in Section 8.4

8.3 Variable Impedance Control

Impedance control (Hogan, 1985) is one of the prominent interaction control approaches, where the relationship between the manipulator position and contact force is related through tunable impedance parameters. Impedance control can be achieved from two directions: passively through hardware design or actively through the controller. In this paper, we focus on the problem of active impedance control. In classical impedance control, the impedance properties (i.e. stiffness, damping, and mass matrices) are considered fixed throughout the task. Recent approaches employ a variable impedance control law with the aim to provide higher performance, efficiency, and human safety (Ganesh et al., 2012; Kronander and Billard, 2013; Mitrovic et al., 2011).

Stability analysis of variable impedance control is non-trivial and is only evaluated in a few works. In (Ferraguti et al., 2013), a passivity-based approach is proposed to ensure stability of a time-varying impedance controller. This method evaluates the power balance of the robot and ensures that the amount of energy pumped into the system is always less than the dissipated energy, and thus the system remains passive. In (Lee and Buss, 2008), stability analysis of a force tracking impedance controller is presented. Ganesh et al. (2012) describe an adaptive approach to guarantee stability of their variable impedance controller. All these approaches analyze stability for cases where the reference trajectory is fixed throughout the motion.

8.4 Robot Learning

Manual programming of robot motions often requires a large amount of engineering knowledge about both the task and the robot and, above that, it can become particularly non-intuitive when dealing with high degrees of freedom robotic systems or fulfilling requirements of complex tasks. Learning-based approaches appear as a promising route to automate this phase by having the robot actually learn how to perform a desired task. Imitation learning and reinforcement learning are two well-known techniques to learn robot motions. Robot learning has been used in various applications. In this paper we solely review the works that are related to either learning potential fields, primitives, or (variable) impedance controllers.

Several approaches consider the problem of learning impedance parameters. Cohen and Flash (1991) propose an impedance learning strategy based on associative search network for tasks in contact. They applied this approach on the task of wiping a surface with unknown geometry. Kim et al. (2010) consider an impedance control strategy based on a human motor control theory, the equilibrium point control model, and use reinforcement learning to optimize the performance of the task in contact. Li et al. (2014) investigate impedance control for object manipulation at two levels of impedance for stable grasping and impedance for object manipulation. They propose an imitation learning algorithm to learning the desired object's manipulation stiffness from human demonstration. Howard et al. (2013) present an inverse optimal control to extract the objective of a demonstrated behavior for a given task, and then use a reinforcement learning technique to transfer the user impedance strategy to a variable impedance robot.

To provide realtime adaptation to changes in dynamic environments, a new body of research is directed at using the combination of variable impedance control with movement primitives. These works are also closest to this paper as they deal with both motion generation and impedance control. Calinon et al. (2010b) propose an imitation learning approach to model robot motions with dynamical systems and formulate the stiffness matrix to be inversely proportional to the observed covariance in the demonstrations. A similar formulation is used in (Kormushev et al., 2010) with the difference that the impedance parameters are estimated through a reinforcement learning approach (as opposed to learning from demonstrations). Stulp et al. (2012) and Buchli et al. (2011) also take a reinforcement learning approach to estimate the variable impedance control law for a given task.

Despite the importance of ensuring stability of the system (especially when robots must perform tasks in dynamically changing environments or in the vicinity of humans), existing works either do not perform such stability analysis (Calinon et al., 2010b; Kronander and Billard, 2013) or only evaluate it numerically along a particular desired trajectory using reinforcement learning techniques (i.e. considering local stability) (Buchli et al., 2011; Kormushev et al., 2010; Stulp et al., 2012).

The research in learning potential fields has been mostly focused on mobile navigation. Gomez et al. (2012) propose an imitation learning algorithm to build a potential function, represented by Fast Marching Square, from a set of human demonstrations. They evaluate their approach on a set of 2D examples, with special focus on mobile navigation. Pipe (2000) considers the problem of autonomously acquiring a 2D potential field with the goal to represent a cognitive map. They use radial basis function neural networks to store the explored map. Howard et al. (2010) present an approach to learn a potential function from a set of constrained motions with the aim to generalize across different constraints. The main difference between the proposed approach and these works is that UMIC can additionally encapsulate explicit information on the robot's interaction properties, i.e. stiffness and damping matrices, in the potential function. In other words, our approach presents a potential field that could account for both the robot motion and its interaction with the environment. Apart from this major difference, UMIC could encode the task dynamics (i.e. following a specific velocity profile) by learning damping parameters; an important feature which allows to generate more-natural looking motions that truly replicates the desired motion (as shown in the demonstration). Furthermore, the proposed controller allows to represent complex-shaped potential functions without any need to discretize the state space. The computational complexity of UMIC scales linearly with dimension, hence can be scaled to high-dimensional problems.

In our previous work (Khansari-Zadeh et al., 2014) we propose a controller that integrates feedback motion generation and variable impedance control in one control unit with guaranteed stability. This work extends our previous work in several aspects: 1) The control policy can be easily learned/modified based on a few demonstrations, 2) It is non-parametric and can model a considerably wider set of robot motions, and 3) It allows explicitly changing stiffness without modifying the motion, and vice versa.

9 Summary and Conclusion

In this paper we presented a novel time-invariant controller, called Unified Motion and variable Impedance Control (UMIC). By integrating motion generation and variable impedance control, our method creates a safe and compliant robot controller without sacrificing accuracy in reaching the final point. In addition, we proposed a learning algorithm based on solving two convex-quadratic optimizations to build an estimate of UMIC from human demonstrations. UMIC is non-parametric and is capable of encoding complex behaviors. Furthermore, it has guaranteed stability in contact with passive environments, and is computationally fast for implementation on robots.

We evaluated the performance of our learning algorithm on a library of human handwriting motions and on the 7-DOF KUKA LWR arm. We showed that our controller is capable of encoding complex patterns as long as they can be disambiguately defined in the state space. In the robot experiment we considered a worst-case scenario in which the robot can only sense the target position, hence no obstacle avoidance or evasive strategy can be performed to avoid collision with human and other objects in the environment. Through this experiment we highlighted the capability of our controller to safely execute the task and precisely reach the target point if it is feasible. Furthermore, we showed that despite unifying motion generation and variable impedance control, our controller still allows customizing each behavior (i.e. motion or interaction) separately.

We also provided a comparison between UMIC and two commonly used active variable impedance control techniques, highlighting the pros and cons of our approaches compared to these methods. Our approach outperforms these techniques with the ability to adapt to changing environments as well as providing a better safety margin in collision with unperceived objects. Future works are directed at extending our controller to perform tooling and assembly tasks, where additional constraints need to be considered to successfully execute the task.

Acknowledgments

Mohammad Khansari is supported by the Swiss National Science Foundation.

References

A. Billard, S. Calinon, R. Dillmann, and S. Schaal. *Handbook of Robotics*, chapter Robot Programming by Demonstration, pages 1371–1394. Springer Berlin Heidelberg, 2008.

Oliver Brock, James Kuffner, and Jing Xiao. *Handbook of Robotics*, chapter Motion for Manipulation Tasks, pages 615–645. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-30301-5_6.

Jonas Buchli, Freek Stulp, Evangelos Theodorou, and Stefan Schaal. Learning variable impedance control. *The International Journal of Robotics Research*, 30(7):820–833, 2011.

S. Calinon, F. D'halluin, E.L. Sauser, D.G. Caldwell, and A.G. Billard. Learning and reproduction of gestures by imitation: An approach based on Hidden Markov Model and Gaussian Mixture Regression. *IEEE Robotics and Automation Magazine*, 17:2:44–54, 2010a.

S. Calinon, I. Sardellitti, and D.G. Caldwell. Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In *proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 249–254, 2010b.

S. Calinon, A. Pistillo, and D. G. Caldwell. Encoding the time and space constraints of a task in explicit-duration hidden Markov model. In *proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3413–3418, 2011.

M. Cohen and T. Flash. Learning impedance parameters for robot control using an associative search network. *Robotics and Automation, IEEE Transactions on*, 7(3): 382–390, 1991. doi: 10.1109/70.88148.

F. Ferraguti, C. Secchi, and C. Fantuzzi. A tank-based approach to impedance control with variable stiffness. In *proc IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4948–4953, May 2013.

G. Ganesh, N. Jarrasse, S. Haddadin, A. Albu-Schaeffer, and E. Burdet. A versatile biomimetic controller for contact tooling and haptic exploration. In *proc IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3329–3334, May 2012.

J.V. Gomez, D. Alvarez, S. Garrido, and L. Moreno. Kinesthetic teaching via fast marching square. In *proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1305–1310, Oct 2012.

Elena Gribovskaya, Seyed Mohammad Khansari-Zadeh, and Aude Billard. Learning Nonlinear Multivariate Dynamics of Motion in Robotic Manipulators. *The International Journal of Robotics Research*, 30:1–37, 2010.

S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger. Collision detection and reaction: A contribution to safe physical human-robot interaction. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3356–3363, 2008.

N. Hogan. Impedance control: an approach to manipulation. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107, 1985.

N Hogan and S P Buerger. *Robotics and automation handbook*, chapter Impedance and Interaction Control. Boca Raton, FL: CRC, 2005.

M. Howard, D.J. Braun, and S. Vijayakumar. Transferring human impedance behavior to heterogeneous variable impedance actuators. *IEEE Transactions on Robotics*, 29 (4):847–862, 2013. doi: 10.1109/TRO.2013.2256311.

Matthew Howard, Stefan Klanke, Michael Gienger, Christian Goerick, and Sethu Vijayakumar. Methods for learning control policies from variable-constraint demonstrations. In *From Motor Learning to Interaction Learning in Robots*, volume 264, pages 253–291. Springer Berlin Heidelberg, 2010.

A. J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1398–1403, 2002.

L. E. Kavraki and Steven M. LaValle. *Handbook of Robotics*, incollection Motion Planning, pages 109–131. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-30301-5_6.

Seyed Mohammad Khansari-Zadeh. Lasa human handwriting library, 2011. Available online at *http://lasa.epfl.ch/khansari/LASA_Handwriting_Dataset.zip*.

Seyed Mohammad Khansari-Zadeh. *A Dynamical System-based Approach to Modeling Stable Robot Control Policies via Imitation Learning*. Phd thesis, cole Polytechnique Fdrale de Lausanne, November 2012. URL *http://infoscience.epfl.ch/record/182663*.

Seyed Mohammad Khansari-Zadeh and A. Billard. Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Trans. on Robotics*, 27(5):943–957, 2011. ISSN 1552-3098. doi: 10.1109/TRO.2011.2159412.

Seyed Mohammad Khansari-Zadeh and A. Billard. A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32:433–454, 2012. ISSN 0929-5593.

Seyed Mohammad Khansari-Zadeh and A. Billard. Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6):752–765, 2014.

Seyed Mohammad Khansari-Zadeh, A. Lemme, Y. Meirovitch, B. Schrauwen, M.A. Giese, J. Steil, A.J. Ijspeert, and A. Billard. Benchmarking of state-of-the-art algorithms in generating human-like robot reaching motions. Workshop at the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids), 2013. Online at *http://www.amarsi-project.eu/news/humanoids-2013-workshop*.

Seyed Mohammad Khansari-Zadeh, K. Kronander, and A. Billard. Modeling robot discrete movements with state-varying stiffness and damping: A framework for integrated motion generation and impedance control. In *proc. of Robotics: Science and Systems X (RSS 2014)*, Berkeley, California, 2014.

O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5: 90–98, 1986.

O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, 3:43–

53, 1987.

Oussama Khatib. Inertial properties in robotic manipulation: An object-level framework. *The International Journal of Robotics Research*, 14(1):19–36, 1995.

Oussama Khatib, Luis Sentis, and Jae-Heung Park. A Unified Framework for Whole-Body Humanoid Robot Control with Multiple Constraints and Contacts. In *European Robotics Symposium 2008*, volume 44 of *Springer Tracts in Advanced Robotics*, pages 303–312. Springer Berlin Heidelberg, 2008.

Byungchan Kim, Jooyoung Park, Shinsuk Park, and Sungchul Kang. Impedance learning for robotic contact tasks using natural actor-critic algorithm. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(2):433–443, 2010. doi: 10.1109/TSMCB.2009.2026289.

J.-O. Kim and P.K. Khosla. Real-time obstacle avoidance using harmonic potential functions. *IEEE Trans. on Robotics and Automation*, 8(3):338–349, 1992.

Y. Kishi, Y. Yamada, and K. Yokoyama. The role of joint stiffness enhancing collision reaction performance of collaborative robot manipulators. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 376–381, 2012.

Daniel Koditschek. Robot planning and control via potential functions. pages 349–367, 1989.

Petar Kormushev, S. Calinon, and D. G. Caldwell. Robot Motor Skill Coordination with EM-based Reinforcement Learning. In *proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 3232–3237, Taipei, Taiwan, October 2010.

K. Kronander and A. Billard. Learning compliant manipulation through kinesthetic and tactile human-robot interaction. *IEEE Transactions on Haptics*, 7(3):367–380, 2013.

K. Lee and M. Buss. Force tracking impedance control with variable target stiffness. In *proc. of Int. Federation of Automatic Control World Congress*, pages 6751–6756, 2008.

Miao Li, Hang Yin, K. Tahara, and A. Billard. Learning object-level impedance control for robust grasping and dexterous manipulation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6784–6791, 2014.

Jacob Mattingley and Stephen Boyd. Cvxgen: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012. ISSN 1389-4420.

Djordje Mitrovic, Stefan Klanke, and Sethu Vijayakumar. Learning impedance control of antagonistic systems based on stochastic optimization principles. *The International Journal of Robotics Research*, 30(5):556–573, 2011.

K. Muelling, J. Kober, O. Kroemer, and J. Peters. Learning to Select and Generalize Striking Movements in Robot Table Tennis. *International Journal of Robotics Research*, 32:263–279, 2013.

Christian Ott. *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, volume 49 of *Springer Tracts in Advanced Robotics*. Springer Tracts in Advanced Robotics, 2008.

Anthony G. Pipe. An architecture for learning "potential field" cognitive maps with an application to mobile robotics. *Adaptive Behavior*, 8(2):173–203, 2000. doi: 10.1177/105971230000800205.

E. Rimon and D.E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, oct 1992. doi: 10.1109/70.163777.

Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.

B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer, 2009.

J.J.E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall, 1991.

F. Stulp, J. Buchli, A. Ellmer, M. Mistry, E. Theodorou, and S. Schaal. Model-free Reinforcement Learning of Impedance Control in Stochastic Environments. *IEEE Transactions on Autonomous Mental Development*, 4(4):330–341, 2012.

A. Ude, A. Gams, T. Asfour, and J. Morimoto. Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives. *IEEE Trans. on Robotics*, 26(5):800–815, 2010.

L. Villani and J. De Schutter. *Handbook of Robotics*, chapter Force Control, pages 161–185. Springer Berlin Heidelberg, 2008.

S. Wolf and G. Hirzinger. A new variable stiffness design: Matching requirements of the next robot generation. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1741–1746, 2008. doi: 10.1109/ROBOT.2008.4543452.

M. Zinn, O. Khatib, B. Roth, and J.K. Salisbury. Playing it safe [human-friendly robots]. *IEEE Robotics Automation Magazine*, 11(2):12–21, 2004. ISSN 1070-9932. doi: 10.1109/MRA.2004.1310938.

Appendices

A Coefficients of the Optimization Function

As described in Section 4, the optimization problem given by Eq. (13) can be transformed into the well-known form:

$$\min_{\boldsymbol{\Theta}} J(\boldsymbol{\Theta}) = \frac{1}{2}\boldsymbol{\Theta}^T \boldsymbol{H}\boldsymbol{\Theta} + \boldsymbol{h}^T\boldsymbol{\Theta} + h_0 \tag{20a}$$

subject to

$$\boldsymbol{\mathcal{C}\Theta} \leq \mathbf{0} \tag{20b}$$

$$\boldsymbol{\mathcal{G}\Theta} = \boldsymbol{g} \tag{20c}$$

where $\boldsymbol{H} \in \mathbb{R}^{\mathcal{T}\times\mathcal{T}}$ is a symmetric positive semi-definite matrix[5], $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{\mathcal{T}\times\mathcal{T}}$ and $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{d\times\mathcal{T}}$ are full rank matrices, $\boldsymbol{h} \in \mathbb{R}^{\mathcal{T}}$, $\boldsymbol{g} \in \mathbb{R}^{d}$, and h_0 is a scalar independent of $\boldsymbol{\Theta}$.

In this appendix, we provide the formulation to compute the coefficients of this quadratic optimization function as given by Eq. (20). The first step to reach our goal is to transform the term $-\nabla\Phi(\boldsymbol{\xi}^i;\boldsymbol{\Theta})$ into the form $\boldsymbol{A}^i\boldsymbol{\Theta}+\boldsymbol{b}^i$, where $\boldsymbol{A}^i \in \mathbb{R}^{d\times\mathcal{T}}$ and $\boldsymbol{b}^i \in \mathbb{R}^{d}$. Let us define:

$$\boldsymbol{\eta}^{ik} = \frac{\tilde{\omega}^k(\boldsymbol{\xi}^i)}{(\sigma^k)^2}(\boldsymbol{\xi}^i - \boldsymbol{\xi}^k) \tag{21a}$$

$$\boldsymbol{\rho}^i = \sum_{k=1}^{\mathcal{T}} \boldsymbol{\eta}^{ik} \tag{21b}$$

$$\upsilon^{ik} = \frac{1}{2}(\boldsymbol{\xi}^i - \boldsymbol{\xi}^k)^T \boldsymbol{S}^k (\boldsymbol{\xi}^i - \boldsymbol{\xi}^k) \tag{21c}$$

then we have:

$$\boldsymbol{A}^i = \begin{bmatrix}\boldsymbol{\eta}^{i1} - \tilde{\omega}^1(\boldsymbol{\xi}^i)\boldsymbol{\rho}^i & \cdots & \boldsymbol{\eta}^{i\mathcal{T}} - \tilde{\omega}^{\mathcal{T}}(\boldsymbol{\xi}^i)\boldsymbol{\rho}^i\end{bmatrix} \tag{22a}$$

$$\boldsymbol{b}^i = \sum_{k=1}^{\mathcal{T}} \upsilon^{ik}\boldsymbol{\eta}^{ik} - \tilde{\omega}^k(\boldsymbol{\xi}^i)\Big(\upsilon^{ik}\boldsymbol{\rho}^i + \boldsymbol{S}^k(\boldsymbol{\xi}^i - \boldsymbol{\xi}^k)\Big) \tag{22b}$$

The desired gradient for the i-th data point is also given by $\boldsymbol{\gamma}^i$ (see Section 4). To account for all the points in the data set, we concatenate the matrices $\boldsymbol{A}^i$, vectors $\boldsymbol{b}^i$ and $\boldsymbol{\gamma}^i$ into a bigger matrix $\boldsymbol{A} \in \mathbb{R}^{\mathcal{T}d\times\mathcal{T}}$ and vectors $\boldsymbol{b} \in \mathbb{R}^{\mathcal{T}d}$ and $\boldsymbol{\gamma} \in \mathbb{R}^{\mathcal{T}d}$:

$$\boldsymbol{A} = \begin{bmatrix}(\boldsymbol{A}^1)^T & \cdots & (\boldsymbol{A}^{\mathcal{T}})^T\end{bmatrix}^T \tag{23a}$$

$$\boldsymbol{b} = \begin{bmatrix}\boldsymbol{b}^1 & \cdots & \boldsymbol{b}^{\mathcal{T}}\end{bmatrix}^T \tag{23b}$$

$$\boldsymbol{\gamma} = \begin{bmatrix}\boldsymbol{\gamma}^1 & \cdots & \boldsymbol{\gamma}^{\mathcal{T}}\end{bmatrix}^T \tag{23c}$$

Then we could obtain the equivalent of Eq. (13a) in the quadratic form $\frac{1}{2}\boldsymbol{\Theta}^T\boldsymbol{H}\boldsymbol{\Theta} + \boldsymbol{h}^T\boldsymbol{\Theta} + h_0$:

$$\boldsymbol{H} = 2\boldsymbol{A}^T\boldsymbol{A} \tag{24a}$$

$$\boldsymbol{h} = 2\boldsymbol{A}^T(\boldsymbol{b} - \boldsymbol{\gamma}) \tag{24b}$$

$$h_0 = (\boldsymbol{b} - \boldsymbol{\gamma})^T(\boldsymbol{b} - \boldsymbol{\gamma}) \tag{24c}$$

The matrix $\boldsymbol{\mathcal{C}}$ in Eq. (20b) can be computed by:

$$\boldsymbol{\mathcal{C}}^{ij} = \begin{cases} -1 & i = j \\ 1 & i = j-1,\ i \notin \Omega \\ 0 & otherwise \end{cases} \tag{25}$$

where $\boldsymbol{\mathcal{C}}^{ij}$ refers to the component in the i-th row and j-th column of $\boldsymbol{\mathcal{C}}$, and Ω is the set of indices that corresponds to the last point of each demonstration trajectory.

The matrix $\boldsymbol{\mathcal{G}}$ and vector $\boldsymbol{g}$ in Eq. (20c) are in fact equal to $\boldsymbol{A}^{\mathcal{T}}$ and $-\boldsymbol{b}^{\mathcal{T}}$, respectively. This is due to the fact that the last point of each demonstration trajectory is in fact the target point, hence $\boldsymbol{\xi}^{\mathcal{T}} = \boldsymbol{\xi}^*$. Note that as we have N trajectories, we could use any of the N available $\boldsymbol{A}^i$ and $\boldsymbol{b}^i$, $i \in \Omega$ to obtain $\boldsymbol{\mathcal{G}}$ and vector $\boldsymbol{g}$. Here for simplicity we use the last point of the last trajectory which by construction has the index $\mathcal{T}$.

[5] Note that depending on the value of σ^i, the matrix $\boldsymbol{H}$ may have one or more small eigenvalues ($< 10^{-10}$). Their associated eigenvectors correspond to indifference to shift in $\boldsymbol{\Theta}$; hence there could be infinite equally optimal solutions. Among these solutions, we choose the one with minimum norm.

B Stability Proof

For a manipulator with d generalized degrees of freedom $\boldsymbol{\xi} \in \mathbb{R}^d$, the robot dynamics (whether in operational or joint space) can be represented by (Khatib, 1995; Ott, 2008; Siciliano et al., 2009):

$$\boldsymbol{M}(\boldsymbol{\xi})\ddot{\boldsymbol{\xi}} + \boldsymbol{C}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}})\dot{\boldsymbol{\xi}} + \boldsymbol{g}(\boldsymbol{\xi}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \tag{26}$$

where $\boldsymbol{M}(\boldsymbol{\xi}) \in \mathbb{R}^{d\times d}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}}) \in \mathbb{R}^{d\times d}$ is the Coriolis/centrifugal matrix, $\boldsymbol{g}(\boldsymbol{\xi})$ is the gravitational force, $\boldsymbol{\tau}$ represents the actuators generalized force, and $\boldsymbol{\tau}_{ext}$ is the external generalized force applied to the robot by the environment.

Similarly to an impedance controller, the actuators generalized force in our control setting is composed of two terms: the gravitational term $\boldsymbol{g}(\boldsymbol{\xi})$ to compensate for the weight of the robot and the controller term $\boldsymbol{\tau}_c$ to perform the task:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_c + \boldsymbol{g}(\boldsymbol{\xi}) \tag{27}$$

To verify stability of UMIC, we use the following definition of passivity, taken from (Slotine and Li, 1991):

Definition 1 A system with input effort v and output flow y is passive if it satisfies:

$$\dot{V} = v^T y - n \tag{28}$$

for some lower bounded scalar functions V and $n \geq 0$.

In our setting, the effort is $\boldsymbol{\tau}_{ext}$ and the flow is $\dot{\boldsymbol{\xi}}$. To ensure passivity/stability of our controller, we define the following candidate Lyapunov function:

$$V(\boldsymbol{\xi},\dot{\boldsymbol{\xi}}) = \Phi(\boldsymbol{\xi}) + \frac{1}{2}\dot{\boldsymbol{\xi}}^T\boldsymbol{M}(\boldsymbol{\xi})\dot{\boldsymbol{\xi}} \tag{29}$$

Taking the time-derivative of $V(\boldsymbol{\xi},\dot{\boldsymbol{\xi}})$ yields:

$$\dot{V}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}}) = \dot{\boldsymbol{\xi}}^T\nabla\Phi(\boldsymbol{\xi}) + \dot{\boldsymbol{\xi}}^T\boldsymbol{M}(\boldsymbol{\xi})\ddot{\boldsymbol{\xi}} + \frac{1}{2}\dot{\boldsymbol{\xi}}^T\dot{\boldsymbol{M}}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}})\dot{\boldsymbol{\xi}} \tag{30}$$

The term $\boldsymbol{M}(\boldsymbol{\xi})\ddot{\boldsymbol{\xi}}$ can be obtained by rearranging Eq. (26):

$$\boldsymbol{M}(\boldsymbol{\xi})\ddot{\boldsymbol{\xi}} = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} - \boldsymbol{C}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}})\dot{\boldsymbol{\xi}} - \boldsymbol{g}(\boldsymbol{\xi}) \tag{31}$$

Furthermore from Eqs. (1) and (27), we have $\boldsymbol{\tau} = \boldsymbol{\tau}_c + \boldsymbol{g}(\boldsymbol{\xi}) = -\nabla\Phi(\boldsymbol{\xi}) - \boldsymbol{\Psi}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}}) + \boldsymbol{g}(\boldsymbol{\xi})$. Considering this and the skew-symmetric property $\dot{\boldsymbol{M}}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}}) - 2\boldsymbol{C}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}})$, Eq. (31) can be simplified to:

$$\begin{aligned}\dot{V}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}}) = &-\dot{\boldsymbol{\xi}}^T\boldsymbol{\Psi}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}}) - \dot{\boldsymbol{\xi}}^T\boldsymbol{\tau}_{ext} \\ &\underbrace{-\dot{\boldsymbol{\xi}}^T\Big(\sum_i \tilde{\omega}^i(\boldsymbol{\xi})\boldsymbol{D}^i\Big)\dot{\boldsymbol{\xi}}}_{\boldsymbol{n}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}})\geq 0} + \dot{\boldsymbol{\xi}}^T\boldsymbol{\tau}_{ext} \\ &= -\boldsymbol{n}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}}) + \dot{\boldsymbol{\xi}}^T\boldsymbol{\tau}_{ext}\end{aligned} \tag{32}$$

Note that $\boldsymbol{n}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}}) \geq 0$ because D^i are positive definite matrices and by construction $\tilde{\omega}^i(\boldsymbol{\xi}) > 0$. Hence, following Definition 1, UMIC yields a passive map from the external generalized force to the velocity of the manipulator.