ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/339627382

Force Adaptation in Contact Tasks with Dynamical Systems

Conference Paper · March 2020

CITATIONS
0

READS
136

4 authors:



Walid Amanhoud
École Polytechnique Fédérale de Lausanne
4 PUBLICATIONS **6** CITATIONS
SEE PROFILE



Mahdi Khoramshahi
École Polytechnique Fédérale de Lausanne
48 PUBLICATIONS **256** CITATIONS
SEE PROFILE



Maxime Bonnesoeur
École Polytechnique Fédérale de Lausanne
1 PUBLICATION **0** CITATIONS
SEE PROFILE



Aude Billard
École Polytechnique Fédérale de Lausanne
808 PUBLICATIONS **14,825** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

CoWriter View project

SecondHands View project

All content following this page was uploaded by Mahdi Khoramshahi on 02 March 2020.

The user has requested enhancement of the downloaded file.

Force Adaptation in Contact Tasks with Dynamical Systems

Walid Amanhoud[1], Mahdi Khoramshahi[1], Maxime Bonnesoeur[1], and Aude Billard[1]

***Abstract*— In many tasks such as finishing operations, achieving accurate force tracking is essential. However, uncertainties in the robot dynamics and the environment limit the force tracking accuracy. Learning a compensation model for these uncertainties to reduce the force error is an effective approach to overcome this limitation. However, this approach requires an adaptive and robust framework for motion and force generation. In this paper, we use the time-invariant Dynamical System (DS) framework for force adaptation in contact tasks. We propose to improve force tracking accuracy through online adaptation of a state-dependent force correction model encoded with Radial Basis Functions (RBFs). We evaluate our method with a KUKA LWR IV+ robotic arm. We show its efficiency to reduce the force error to a negligible amount with different target forces and robot velocities. Furthermore, we study the effect of the hyper-parameters and provide a guideline for their selection. We showcase a collaborative cleaning task with a human by integrating our method to previous works to achieve force, motion, and task adaptation at the same time. Thereby, we highlight the benefits of using adaptive force control in real-world environments where we need reactive and adaptive behaviours in response to interactions with the environment.**

***Index Terms*— Force Control, Compliance and Impedance Control, Physical Human-Robot Interaction.**

I. Introduction

Many tasks require robots to enter in contact with arbitrary surfaces, move on them while applying desired contact forces. For finishing operations such as polishing or grinding, the contact force exerted on the surface is a key process variable, as the amount of force applied directly affects the material removal rate [1]. Besides following accurately the desired motion and force profile, the robots should also be compliant to deal with inaccurate modelling of the surfaces (e.g., stiffness and location) and real-time disturbances such as those introduced by humans or unexpected changes in the environment [2]. This requires the development of suitable control strategies to regulate the robot movement and interaction forces with the environment.

Most of the current strategies are based on impedance control [3] which regulates the dynamic relationship between interaction forces and motion deviations through a mass-spring-damper system. Impedance controllers provide a compliant behaviour in all phases of a contact task (e.g, non-contact, transition and contact) but are limited in their force tracking ability, mainly from partial knowledge of the environment. To cope with this limitation, two distinct strategies are usually employed in the literature: impedance and set-point adaptation. Impedance adaptation adjusts the impedance parameters online (e.g., inertia, damping, and stiffness) to improve tracking in response to force, position, or velocity measurements [4]–[7]. Set-point adaptation approach improves force tracking by adjusting the impedance set-point (e.g., the reference position) based on force tracking error or on real-time estimation of the environment's change in stiffness [8]–[12]. To compensate the tracking error due to environment uncertainties, many works have also focused on learning control in an iterative manner [13]. Iterative Learning Control (ILC) aims to improve the task tracking performance of a system by learning a feed-forward compensation signal based on signals obtained from previous execution cycles of the tasks. This is particularly useful for repetitive tasks such as polishing; see [14]–[17] as examples where ILC is used to correct a reference trajectory profile to maintain a desired contact force.

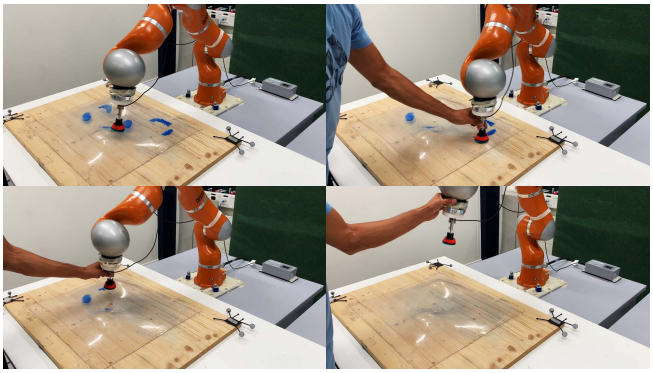


Fig. 1: A robot arm cleans a non-flat surface in collaboration with a human. It moves circularly on the surface while accurately generating contact forces (top-left). Through the interaction with the surface, the robot can adapt the desired force profile to reduce the force error. Through the interaction with the human, the robot adapts the cleaning area (top-right and bottom-left) and compliantly retracts from the surface (bottom-right).

Even though the aforementioned works provide accurate motion and force tracking performance, their robustness to real-time disturbances is either not mentioned or limited to small disturbances, such as in the surface location or in the environment stiffness. They do not address large disturbances such as those introduced by human interactions. For instance, let us consider the scenario in Fig. 1 where a robot cleans a non-flat surface in collaboration with a human operator. At any moment, the human might want to change the cleaning area by physically interacting with the robot or stop the cleaning by breaking the contact with the surface. In such situations, the robot should not only damp the disturbances using the impedance control (which is only effective for small disturbances) but also "react" and "adapt" its behaviour on the fly. To properly react and adapt, the robot is required to re-plan the execution of the task from the disturbed

[1]LASA Laboratory, Swiss Federal School of Technology in Lausanne - EPFL, Switzerland {walid.amanhoud, mahdi.khoramshahi, maxime.bonnesoeur,aude.billard}@epfl.ch

state and modify the task based on the interaction with the environment. These reactivity and adaptability need to be continuous, smooth, and robust toward human highly dynamic behaviour and other uncertainties in the environment. Representing tasks with time-indexed references for position and force profiles is the main drawback in current approaches in achieving fast reactivity toward large disturbances; see [11], [12], [16], [17] as examples where a time-dependent representation of the task is used. In contrast, in a state-dependent and time-invariant task representation, interactions with the environment can be captured by changes in the robot's state which can be used in the modification and re-planning of the task [18].

In this paper, we exploit time-invariant control through the Dynamical System (DS) method to provide fast reactivity and on-the-fly re-planning of trajectories [19]. While for a long-time, such approach was used to control only trajectories in free space, recent work showed that it could be extended to control for impedance when making contact with objects [20], [21] and when performing physical interaction with humans [22]. These works could however only control for stiffness at contact but not for the amount of force applied. Recently, in [23], we extended this framework to control explicitly for the amount of forces when coming in contact with the object. The strategy is based on local modulation of the robot's nominal motion to generate contact forces when the robot is close to the target surface. With this strategy, a robot can perform contact tasks and react to human interactions (e.g., stopping the robot, breaking the contact, and moving the robot arbitrarily), or unexpected changes in the environment (e.g., the position and orientation of the surface/object). However, the generation of the desired contact force is done in open loop without any force feedback. It hence assumes that the robot's dynamics (gravity, inertia, etc) are compensated for and that the stiffness of the contact surface is known. This results, in practice, in errors in the force produced at the end-point. Errors come from uncertainties about the contact surface (e.g, location of the surface, normal of the surface, friction), robot model and measurement noises. Most of these uncertainties are however structural and can be modelled or corrected for. This is particularly true for repetitive tasks where the robot needs to repeat the same motion while maintaining a desired force profile. The robot can learn and adapt online to these errors through the interaction with the surface. In this work, we extend the approach presented in [23] to deal with uncertainties and improve the force tracking performance through on-line learning of a state-dependent force correction model. The advantage of learning a state-dependent force compensation is that the online adaptation can be deactivated once the error has been sufficiently reduced and the learned profile can be re-used later without the need to re-learn. We present our method in section II, we evaluate it in section III, and we conclude with a discussion about the method and results obtained in section IV.

II. METHOD

A. Robot's dynamics and control

Let us consider the dynamics of a N degrees of freedom robotic manipulator in the three-dimensional Cartesian space:

$$\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} = \boldsymbol{F_c} - \boldsymbol{F_m} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^3$ denotes the robot's position, $\boldsymbol{M}(\boldsymbol{x}) \in \mathbb{R}^{3\times3}$ the mass matrix, $\boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} \in \mathbb{R}^3$ the centrifugal forces, while $\boldsymbol{F_c} \in \mathbb{R}^3$ and $\boldsymbol{F_m} \in \mathbb{R}^3$ represent the control and measured external forces respectively. Eq. 1 assumes that the gravity forces $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^3$ are already compensated. $\boldsymbol{F_c}$ allows to track a desired velocity profile $\dot{\boldsymbol{x}}_{\boldsymbol{d}} \in \mathbb{R}^3$ and is obtained from the DS-impedance controller in [21]:

$$\boldsymbol{F_c} = \boldsymbol{D}(\boldsymbol{x})(\dot{\boldsymbol{x}}_{\boldsymbol{d}} - \dot{\boldsymbol{x}}) = d_1\dot{\boldsymbol{x}}_{\boldsymbol{d}} - \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} \tag{2}$$

where $\boldsymbol{D}(\boldsymbol{x}) \in \mathbb{R}^{3\times3}$ is a state-varying damping matrix, constructed such that the first eigenvector is aligned with the desired dynamics $\dot{\boldsymbol{x}}_{\boldsymbol{d}}$ with positive eigenvalue $d_1 \in \mathbb{R}^+$. In this work, the DS is applied only to the translation of the end-effector. To control the desired end-effector's orientation we use the axis-angle representation (see [23] for more details). In short, from the measured and desired orientation, the task-space torque is computed using a PD-like control law. The control wrench composed of task-space force (i.e., $\boldsymbol{F_c}$) and torque is then converted into joint torques using the robot's Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{6\times N}$.

B. Force adaptation with dynamical systems

In [23], to achieve the desired motion and force profile with a single DS, we express $\dot{\boldsymbol{x}}_{\boldsymbol{d}}$ as:

$$\dot{\boldsymbol{x}}_{\boldsymbol{d}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{f_n}(\boldsymbol{x}) \tag{3}$$

with $\boldsymbol{f}(\boldsymbol{x})$ the nominal dynamics defining the desired motion, and $\boldsymbol{f_n}(\boldsymbol{x})$ a modulation term that applies only along the direction normal to the surface to generate the desired contact force profile. As a result, $\boldsymbol{F_c}$ becomes:

$$\boldsymbol{F_c} = d_1\boldsymbol{f}(\boldsymbol{x}) + d_1\boldsymbol{f_n}(\boldsymbol{x}) - \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} \tag{4}$$

The first term represents the driving force along the nominal dynamics, the third term is the damping force, while the second term denotes the modulation force along the normal direction to the surface that we design in [23] as:

$$\boldsymbol{f_n}(\boldsymbol{x}) = \frac{F_d(\boldsymbol{x})}{d_1}\boldsymbol{n}(\boldsymbol{x}) \tag{5}$$

where $F_d(\boldsymbol{x}) \in [0, F_{max}]$ is the state-dependent desired force profile with $(F_{max} > 0)$. Concerning the desired motion, we assume that $\boldsymbol{f}(\boldsymbol{x})$ is designed to bring the robot in contact with a surface and move it along the surface. We suppose that the contact surface is non-penetrable and that we have an explicit expression for the normal vector $\boldsymbol{n}(\boldsymbol{x})$ and distance to the surface $\Gamma(\boldsymbol{x})$ at all points in space. The nominal DS should satisfy:

$$\begin{cases} \boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{n}(\boldsymbol{x}) = 0 & \text{in contact} \\ \boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{n}(\boldsymbol{x}) > 0 & \text{in free motion} \end{cases} \tag{6}$$

Such dynamics can be learned from human demonstrations and locally modulated to meet these constraints [18], [24]. The desired orientation is designed to smoothly converge to $\boldsymbol{n}(\boldsymbol{x})$ as the distance to the surface is decreased.

Let us now introduce a state-dependent force correction model $\tilde{F}(\boldsymbol{x}, \boldsymbol{\theta}) \in [-\tilde{F}_{max}, \tilde{F}_{max}]$ with $\tilde{F}_{max} > 0$ where

$\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_K]^T$ is the associated set of parameters of size K. This correction is added to $\boldsymbol{f_n(x)}$ such that:

$$\boldsymbol{f_n(x)} = \frac{F_d(\boldsymbol{x}) + \tilde{F}(\boldsymbol{x}, \boldsymbol{\theta})}{d_1} \boldsymbol{n(x)} \tag{7}$$

Inserting Eq. 4 in Eq. 1, substituting $\boldsymbol{f_n(x)}$ by Eq. 7, projecting everything along $\boldsymbol{n(x)}$, and considering that the robot is in contact (Eq. 6 holds) leads to:

$$\begin{aligned} F_d(\boldsymbol{x}) - \boldsymbol{n(x)}^{\boldsymbol{T}} \boldsymbol{F_m} = \boldsymbol{n(x)}^{\boldsymbol{T}} \big(\boldsymbol{M(x)\ddot{x} + C(x, \dot{x})\dot{x}}\big) \\ - \tilde{F}(\boldsymbol{x}, \boldsymbol{\theta}) + \boldsymbol{n(x)}^{\boldsymbol{T}} \boldsymbol{D(x)\dot{x}} \end{aligned} \tag{8}$$

Our objective is then to minimize the error F_e between the desired and measured contact force along $\boldsymbol{n(x)}$, defined as:

$$F_e = F_d(\boldsymbol{x}) - \boldsymbol{n(x)}^T \boldsymbol{F_m} \tag{9}$$

To this end, one can use the cost function $J = \frac{1}{2} F_e^2$ whose gradient can be used to update the parameters $\boldsymbol{\theta}$:

$$\boldsymbol{\dot{\theta}} = \text{-}\epsilon_r \frac{\partial J}{\partial \boldsymbol{\theta}} = \text{-}\epsilon_r \frac{\partial J}{\partial F_e} \frac{\partial F_e}{\partial \tilde{F}} \frac{\partial \tilde{F}}{\partial \boldsymbol{\theta}} = \epsilon_r F_e \frac{\partial \tilde{F}}{\partial \boldsymbol{\theta}} \tag{10}$$

where $\epsilon_r > 0$ is the adaptation rate.

C. Design of the state-dependent force correction model

To design our state-dependent force correction, we opt for a normalized linear combination of Gaussian Radial Basis kernel functions which is a well-established method in machine learning [25]. Radial Basis Functions (RBF) are often used in applications needing approximation or interpolation especially for their smoothness and nice convergence properties, making them suitable for online adaptation [26]. Thus, we design $\tilde{F}(\boldsymbol{x}, \boldsymbol{\theta})$ as follows:

$$\tilde{F}(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{\sum_{i=1}^{K} \theta_i \phi(\boldsymbol{x - c_i})}{\sum_{j=1}^{K} \phi(\boldsymbol{x - c_j})} \quad \phi(\boldsymbol{x}) = \exp\left(\frac{\text{-}||\boldsymbol{x}||^2}{2\sigma^2}\right) \tag{11}$$

where $\theta_i \in \mathbb{R}$ and $\boldsymbol{c_i} \in \mathbb{R}^3$ respectively denote the weight and center position of gaussian i, while $\sigma > 0$ defines the kernel width for the K gaussians. Given Eq. 11, we can express the gradient $\tilde{F}(\boldsymbol{x}, \boldsymbol{\theta})$ with respect to θ_i, which can be used in Eq. 10 to update $\boldsymbol{\theta}$:

$$\frac{\partial \tilde{F}(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \theta_i} = \frac{\phi(\boldsymbol{x - c_i})}{\sum_{j=1}^{K} \phi(\boldsymbol{x - c_j})} \tag{12}$$

D. Convergence behavior

Let us first assume that the environment dynamics change much slower than the convergence rate. We also assume that the K gaussians are uniformly and disjointly activated over a period T; i.e., each kernel is visited periodically T/Ks. The convergence behaviour of our method can be studied by considering the following form for the force error:

$$F_e = \tilde{F}(\boldsymbol{x}, \boldsymbol{\theta^*}) - \tilde{F}(\boldsymbol{x}, \boldsymbol{\theta}) - \eta(t) \tag{13}$$

where $\boldsymbol{\theta^*}$ is a unique optimal set of parameters modelling the force error, while $\eta(t)$ accounts for unmodelled dynamics. Let us linearize $\tilde{F}(\boldsymbol{x}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ around $\boldsymbol{\theta^*}$:

$$\tilde{F}(\boldsymbol{x}, \boldsymbol{\theta}) = \tilde{F}(\boldsymbol{x}, \boldsymbol{\theta^*}) + \frac{\partial \tilde{F}}{\partial \boldsymbol{\theta}}^T \Big|_{\boldsymbol{\theta = \theta^*}} (\boldsymbol{\theta - \theta^*}) + O(\boldsymbol{\theta}, \boldsymbol{\theta^*}) \tag{14}$$

where $O(\boldsymbol{\theta}, \boldsymbol{\theta^*})$ denotes the high-order terms of the Taylor expansion. Substituting $\tilde{F}(\boldsymbol{x}, \boldsymbol{\theta})$ by Eq. 14 in Eq. 13 gives:

$$F_e = \text{-}\boldsymbol{S(\theta - \theta^*)} + d(t) \tag{15}$$

with $\boldsymbol{S} = \frac{\partial \tilde{F}}{\partial \boldsymbol{\theta}}^T \Big|_{\boldsymbol{\theta = \theta^*}}$ and $d(t) = \text{-}O(\boldsymbol{\theta}, \boldsymbol{\theta^*}) - \eta(t)$. Assuming that the disturbance term $d(t)$ is negligible, the parameters' dynamics in Eq. 10 can be approximated by:

$$\boldsymbol{\dot{\theta}} = \text{-}\epsilon_r \frac{\partial J}{\partial \boldsymbol{\theta}} = \text{-}\epsilon_r \frac{\partial J}{\partial F_e} \frac{\partial F_e}{\partial \boldsymbol{\theta}} \approx \text{-}\epsilon_r \boldsymbol{S^T S(\theta - \theta^*)} \tag{16}$$

The matrix $\boldsymbol{S^T S}$ being positive semi-definite, $\boldsymbol{\theta = \theta^*}$ is a stable equilibrium. Convergence occurs if the force error signal is rich enough, which is often referred as the *Persistent Excitation (PE)* condition in the literature of adaptive control [27]. Here, PE condition for θ_i is satisfied if the area associated with the ith RBF kernel is visited by the robot; i.e., $\partial \tilde{F}(\boldsymbol{x}, \boldsymbol{\theta}) / \partial \theta_i \neq 0$. If convergence of the correction term is proven, stability of the full closed-loop system needs to be considered to ensure stable interaction with the environment. A sufficient condition to achieve stability is to ensure passivity of the whole system [28], [29]. In [23], we achieve passivity through the use of energy tanks [30]. The formulation described in [23] still holds for this work.

III. Experimental evaluations

We evaluate our method in two real-world scenarios. In the first scenario, a robot comes in contact with a non-flat surface, moves on it while generating a desired contact force profile. We assess the ability of the system to improve the force tracking under different conditions and analyze the effect of the choice of hyper-parameters (K and σ) on the granularity and precision of the force modulation. We also show the robustness of our method to real-time disturbances. In the second scenario, we perform a collaborative task with a human where the human asks the robot to clean the surface at different locations. To achieve this, we combine the proposed force adaptation with a mechanism to adapt the attractor of a nominal limit cycle (proposed in our previous works [22], [31]). We show that the force modulation can adapt fast enough to cope with the change in dynamics.

A. Force adaptation on a non-flat surface

In this scenario, we use a 7-DOF robotic arm (KUKA LWR IV+). The robot is equipped with joint torque sensors at the actuators and can be torque-controlled. A 6-axis ATI force-torque sensor is mounted on the end-effector on which a 3D printed finger tool is attached. The rigid non-flat surface is fabricated by deforming a Plexiglas sheet using heat. It is sticked on a wooden plate whose pose is tracked by a motion capture system. Technical details are given in Appendix I.

The robot's behavior is systematically evaluated in a simple task: starting from a fixed initial position, the robot comes in contact with the surface to perform a circular motion (with a fixed center) on the surface while applying a desired contact force. The experiment is repeated 18 times under different choices for hyper-parameters, target force and motion velocity. For each run, force adaptation starts after $9s$ approximately, the experiment lasts $50s$, ϵ_r = 20, while $\tilde{F}_{max}$ is experimentally set to $10N$. To learn $\tilde{F}(\boldsymbol{x}, \boldsymbol{\theta})$, we should define K, σ, and the spatial distribution of the gaussians. The latter mainly depends on the desired robot motion on the surface. We distribute the gaussians uniformly on the surface in a $D \times D$ 2d grid (D = $16cm$) to cover the desired circular motion of radius $5cm$. K and σ need to be picked in relationship to each other to reach accurate and smooth interpolation. To enforce that the gaussians do not overlap

K \ σ	$0.5\sigma_{th}$	σ_{th}	$2\sigma_{th}$
9	**1.45** (3.19)	**0.89** (3.21)	**1.55** (3.23)
100	**0.58** (3.06)	**0.36** (3.19)	**0.79** (3.20)
900	**0.49** (3.19)	**0.69** (3.19)	**1.25** (3.22)

(a)

F_T \ v_0	$0.2m/s$	$0.25m/s$	$0.3m/s$
$10N$	**0.29** (2.69)	**0.32** (3.04)	**0.37** (3.52)
$15N$	**0.31** (2.82)	**0.38** (3.27)	**0.41** (3.46)
$20N$	**0.30** (2.77)	**0.38** (3.09)	**0.48** (3.53)

(b)

TABLE I: Force adaptation on the non-flat surface: RMS values of the force error in N obtained over the last $10s$ of the experiments (in bold). The values in parentheses are the RMS errors obtained in the last $5s$ before starting the adaptation. In Table Ia, different combinations of numbers of RBFs and kernel width are tested while keeping the same target force after contact (F_T = $15N$) and desired robot velocity (v_0 = $0.25m/s$). In Table Ib, the target velocity and contact force are changed while keeping the same adaptation settings (K = 100, σ = σ_{th}).

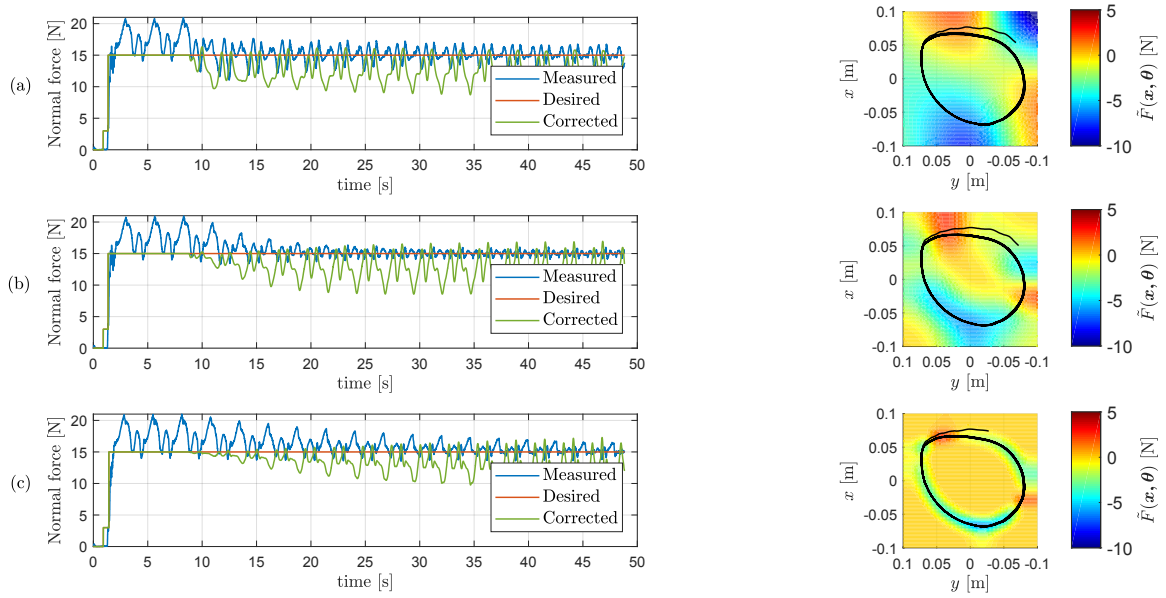


Fig. 2: Force adaptation on the non-flat surface when using different number of RBFs with $\sigma = \sigma_{th}$, F_T = $15N$, and v_0 = $0.25m/s$: (a) K = 9, (b) K = 100, (c) K = 900. On the left side, the measured, desired and corrected desired force profile are depicted. On the right side, the force compensation map learned at the end of the experiments is drawn with respect to the robot's relative position to the attractor, along with the robot path on the surface.

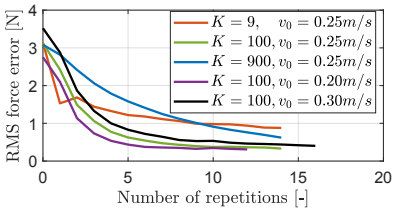


Fig. 3: Force tracking error as a function of the number of repetitions (e.g., circles) with σ = σ_{th}, F_T = $15N$.

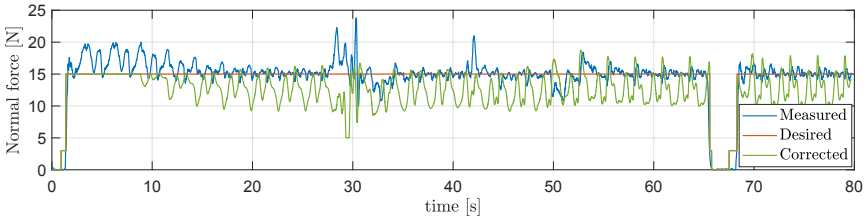


Fig. 4: Force adaptation on the non-flat surface under disturbances with K = 100, σ = σ_{th}, F_T = $15N$, and v_0 = $0.25m/s$.

more than 1 standard deviation, we propose a kernel width of $\sigma_{th} = \frac{D}{\sqrt{K}}$. Finally, the complexity and variability of the unmodeled dynamics need to be considered in selecting K; i.e., higher/lower number of gaussians results in higher/lower approximation power, but higher/lower convergence time.

Table I evaluates the performance of our method by comparing the force tracking error before and after adaptation in terms of Root Mean Square error (RMS). In Table Ia, different combinations of K and σ are tested, while keeping the same target force and desired robot velocity in contact. Before adaptation, the RMS error reaches in average $3.18 \pm 0.06N$. At the end of the experiments, the lowest RMS value ($0.36N$) is obtained by using K = 100 gaussians and setting σ to σ_{th}. We can observe that the RMS error is usually higher when using a significantly smaller or larger σ as expected. Indeed, approximation power of larger kernels suffers from specificity, while small kernels lack generalization/interpolation. Furthermore, with small kernels, high adaptation rates are prone to fluctuations and instabilities. These results also show that lower number of gaussians (K = 9) provides less effective correction. This is illustrated in the measured force profile in Fig. 2a (left) which still has a significant error pattern that cannot be captured by the model. This also translates visually in Fig. 2a (right) where only three distinct regions are visible in the learned force correction map. When increasing K to 100, the approximation of the model improves as shown in Fig. 2b. Higher K is expected to result in lower RMS error. However, it requires more time for convergence as illustrated in the measured force profiles. This is why, higher RMS errors are obtained for K = 900 than for K = 100, as the model parameters are still converging at the end of the experiment. The higher specificity of K = 900 is visible in the force correction map (Fig. 2c (right)) where the

correction is only learned near the motion patterns. In Table Ib, our method is evaluated when targeting different target forces and velocities, while keeping the same adaptation settings. The method shows robustness to desired motion velocity and force where in average, the RMS error is reduced to $0.36 \pm 0.06N$. The lower performance when targeting simultaneously higher desired velocities and forces is expected due to higher frictions. Also, comparing the average results across the two tables ($0.90 \pm 0.42N$ in Table Ia) shows that our method is less sensitive to the desired behavior than the hyper-parameters. This delineates the importance in choosing the hyper-parameters that can be achieved following the proposed guideline. Moreover, Fig. 3 highlights the influence of the number of repetitions (i.e., circles) performed (after starting the adaptation) on the convergence and tracking performance of our method. As expected, for a fixed adaptation rate, higher K or robot velocity need more repetitions to converge. In general, the number of repetitions needed is mainly affected by the adaptation rate, the number of RBFs and the robot velocity.

Finally in Fig. 4, an example of force adaptation under disturbances is provided. Between 27 and $32s$, a human pushes the robot away from the attractor while the robot is moving in contact. The algorithm compensates for the generated force errors by adapting the correction limited by a lower bound. However, once the disturbance disappears, the force error is reduced quickly; i.e. less than $6s$. At $42s$, the human changes the surface inclination, keeps it tilted, and puts it back at $50s$. The correction pattern is adapted quickly (2 to $5s$) to the inclination changes. Our adaptation mechanism only considers the interaction with the surface captured by the force-torque sensor. For instance, when the human takes the robot away from the surface at $65s$ by interacting with the body, the force error is compensated instantaneously once the robot reaches the surface; i.e., the previously adapted model remains unchanged.

B. Collaborative cleaning of a non-flat surface

In this second scenario, we perform a cooperative cleaning task of the non-flat surface with a human. The same setup as in section III-A is used with the exception of the robot tool, replaced with a cleaning pad (see Fig. 1). Here, we highlight the benefits of using dynamical systems to perform contact tasks with a reactive and adaptive behaviour while interacting both with the surface and a human. To this end, we combine:

- The work in [31] to switch across different tasks. For the cleaning of the surface we define two tasks:
  - The homing task (i = 1) defined by $\boldsymbol{f^1(x)} = \boldsymbol{x_{a,h}} - \boldsymbol{x}$ and $F_d^1(\boldsymbol{x}) = 0\ \forall \boldsymbol{x}$, where the robot should reach a fixed attractor above the surface, with $\boldsymbol{x_{a,h}}$ the attractor.
  - The cleaning task (i = 2) defined by $\boldsymbol{f^2(x)}$ and $F_d^2(\boldsymbol{x})$ as in section III-A (see Appendix I).

  From these tasks, we express the adapted nominal DS as:
  $$\boldsymbol{f(x)} = b_1 \boldsymbol{f^1(x)} + b_2 \boldsymbol{f^2(x)} \tag{17}$$
  where $\boldsymbol{b} = [b_1, b_2]^T$ is the belief vector whose update rule ensures that $b_1 + b_2 = 1$ and that one $b_i \rightarrow 1$ in finite time (see [31]). Concerning the overall desired force profile $F_d(\boldsymbol{x})$, we set it to:
  $$F_d(\boldsymbol{x}) = \begin{cases} F_d^i(\boldsymbol{x}) & \text{if } b_i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$
- The work in [22] to adapt the cleaning motion by adjusting the location of the attractor in $\boldsymbol{f^2(x)}$.
- The method in section II-B to adapt the desired force.

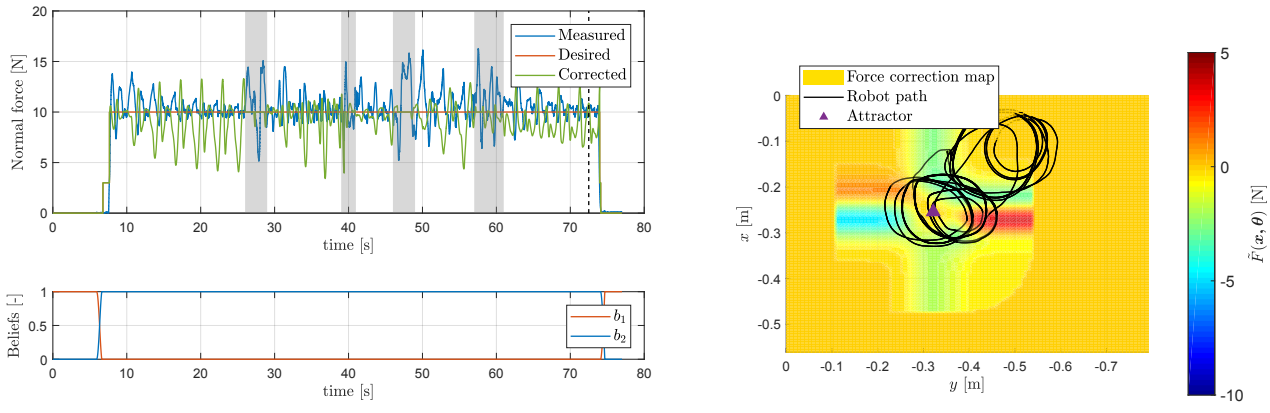
Thus, three adaptation levels occur in this scenario: task, motion, and force. Each of them is activated under simple conditions. Task adaptation is active when the robot is in free motion; i.e., no contact with the surface, or $F_d(\boldsymbol{x})$ = 0. Motion adaptation is activated when the robot is cleaning the surface (i.e., b_2 = 1 and the robot is in contact), and the tangential forces (w.r.t to the surface) perceived at the end-effector exceed a predefined threshold $\varepsilon_{F,t}$. To this end, we assume that the human grabs the robot tool to change the cleaning area and that large forces would result from this interaction. Finally, force adaptation is enabled when the robot is cleaning the surface and the tangential forces remain below $\varepsilon_{F,t}$.

To adapt the desired force profile, we distribute uniformly the gaussians on a grid centered at the current attractor similarly to section III-A; i.e., the gaussians are moving with the adjustable attractor. This avoids using a high number of gaussians to cover the whole surface which suffers from higher computational cost. Because of the adjustable attractor, we reset the weights of the force adaptation when the human is changing the cleaning area. Zero values often serve as a better prior than the previous parameters learned from another area. In line with section III-A, we use K = 100, D = $16cm$, $\sigma = \sigma_{th}$, ϵ_r = 20, and $\tilde{F}_{max} = 10N$. More technical details are provided in Appendix I.

In our collaborative scenario, the robot starts from its home position (b_1 = 1). Following the human physical interaction, the robot switches to the cleaning task (b_2 = 1) and comes in contact with the surface to start cleaning it around a predefined initial attractor. The beliefs' profile illustrated in Fig. 5a (bottom) shows the switching across the two tasks. The activation of the force adaptation occurs after reaching the contact at $8s$; see Fig. 5a (top). The force error quickly reduces to a negligible amount; i.e., RMS error of $0.43N$ between 20 and $25s$. The robot cleans the same area until approximately $25s$ when the human starts to physically interact with the robot with the intention to change the cleaning area. This is illustrated by the large peaks in the measured force profile; see Fig. 5a (top). Fig. 5b illustrates the adaptation of the motion (i.e., the attractor) which occurs several times between 25 and $74s$. During that period, force adaptation is active when the tangential forces are small; i.e., the absence of human interaction. For example, one can see the force correction map and adapted attractor captured at $72.5s$ in Fig. 5b. Finally, at around $74s$, the human stops the cleaning task through physical interaction resulting in the robot retreating; i.e., b_1 = 1.

IV. Summary and Conclusion

In this work, we used dynamical systems for force adaptation in contact tasks. Our method uses online adaptation



(a) (b)

Fig. 5: Collaborative cleaning of the non-flat surface: Force adaptation is illustrated in Fig. 5a top with the measured, desired, and corrected desired force profile. At the bottom, task adaptation is represented by the beliefs' evolution. Motion adaptation is visible in Fig. 5b with the whole robot path projected on the surface's plane during the cleaning task. The gray shared areas in Fig. 5a highlight the interaction periods with the human to adapt the motion. Figure Fig. 5b also shows the attractor and force correction map captured at 72.5s (i.e., vertical dashed line in Fig. 5a).

of a state-dependent force correction model encoded with gaussian RBF kernels. It is particularly suitable for repetitive tasks as shown in section III-A where the robot should repeat the same motion while generating a desired contact force. For non-repetitive tasks which do not require to learn a model (no structural errors), our method could be simply modified to online adaptation of an offset. Our results showed that the force tracking accuracy is significantly improved for different desired motion-force profiles even in the face of real-time disturbances. For the correction to be effective, the hyper-parameters should be set properly; in particular the number of gaussians and kernel width. In our experiments, we distributed the RBFs locally on a small area around the attractor instead of covering the whole surface. The alternative requires a large number of gaussians which increases the computational cost and convergence time. The latter can be reduced using higher adaptation rate which, however, is prone to fluctuations and instabilities. Moreover, the surface used in the experiments remains fairly smooth. With more complex surfaces, a uniform distribution would not be able capture all the surface non-linearities, and increasing globally the number of RBFs is not a proper solution regarding computational cost. One could consider Gaussian Process Regression (GPR) [32] as an alternative to our method which is similar but more generic. Indeed, GPR can model complex non-linear functions and has been applied to learning surfaces [33]. Furthermore, our model only depends on the robot's end-effector position. Such models perform satisfactorily as long as the robot re-visits the same position with similar velocities. Mathematically speaking, non-stationary behaviors with relatively lower dynamics than the adaptation can be compensated. To improve the performance further, one can include other variables such as velocity in the model. However, one has to deal with the curse of dimensionality or find efficient ways to distribute the RBFs in the input space. Overall, our experimental results suggest that many robotic applications can benefit from force adaptation with dynamical systems. As demonstrated in the collaborative cleaning task in section III-B, the time-invariant DS framework provides reactive and adaptive robotic behaviour. This enables robots to perform tasks in uncertain environment where the robot is required to physically interact with humans, objects, and surfaces.

APPENDIX I
TECHNICAL DETAILS

The profile of the non-linear surface is learned with Support Vector Regression (C-SVR) using a gaussian kernel (C = 100, ϵ = 0.01, σ = 0.20) to estimate the normal distance $\Gamma(\boldsymbol{x})$ and vector $\boldsymbol{n}(\boldsymbol{x})$ to the surface at any position in space. These information are learned with respect to a local frame attached to the wooden plate which is tracked by the motion capture system. In both experimental scenarios, the control strategy runs at a frequency of $200Hz$ (e.g., Δ_t = $0.005s$) and the DS-impedance gain d_1 is set to 150.

The circular nominal DS $\boldsymbol{f}(\boldsymbol{x})$ is defined as $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{R}(\boldsymbol{x})\boldsymbol{n}(\boldsymbol{x})v_0$ where $v_0 > 0$ is the target velocity. $\boldsymbol{R}(\boldsymbol{x})$ is a rotation matrix designed to progressively align $\boldsymbol{n}(\boldsymbol{x})$ with circular motion dynamics tangent to the surface, as the robot gets closer to contact. The circular motion has a radius of $5cm$ in both scenarios and a fixed attractor in the first one. The desired force profile $F_d(\boldsymbol{x})$ is defined by:

$$F_d(\boldsymbol{x}) = \begin{cases} F_T & \mu_{F,n} \geq \varepsilon_{F,n} \ \wedge \ \Gamma(\boldsymbol{x}) \leq \varepsilon_\Gamma \\ F_{T,min} & \mu_{F,n} < \varepsilon_{F,n} \ \wedge \ \Gamma(\boldsymbol{x}) \leq \varepsilon_\Gamma \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where $\mu_{F,n}$ is the mean value of the measured normal force over a sliding window of n samples (set to 10) while $\varepsilon_{F,n} \geq 0$ is a force threshold (set to $3N$). $\varepsilon_\Gamma \geq 0$ is the threshold (set to $5cm$) on the normal distance to the surface $\Gamma(\boldsymbol{x})$. $F_{T,min}$ is the target force close to contact (set to $3N$). It ensures that contact with the surface happens and contributes to reduce the impact at contact. F_T is the target force in contact.

For the collaborative task, the contact condition is defined by $\mu_{F,n} \geq \varepsilon_{F,n} \ \wedge \ \Gamma(\boldsymbol{x}) \leq \varepsilon_\Gamma$ while $\varepsilon_{F,t}$ is set to 15 N.

ACKNOWLEDGMENTS

The authors would like to thank for the support from the Hasler Foundation and the European Community's Horizon 2020 through the Research and Innovation programme ICT-2014-1 under grant agreement 643950-SecondHands.

REFERENCES

[1] E. A. Erlbacher, "Force control basics," *Industrial Robot: An International Journal*, vol. 27, no. 1, pp. 20–29, 2000.

[2] T. Lefebvre, J. Xiao, H. Bruyninckx, and G. De Gersem, "Active compliant motion: a survey," *Advanced Robotics*, vol. 19, no. 5, pp. 479–499, 2005.

[3] N. Hogan, "Impedance control: An approach to manipulation," in *American Control Conference, 1984*. IEEE, 1984, pp. 304–313.

[4] R. Ikeura and H. Inooka, "Variable impedance control of a robot for cooperation with a human," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 3. IEEE, 1995, pp. 3097–3102.

[5] K. Lee and M. Buss, "Force tracking impedance control with variable target stiffness," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 6751–6756, 2008.

[6] F. Ferraguti, C. Secchi, and C. Fantuzzi, "A tank-based approach to impedance control with variable stiffness," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4948–4953.

[7] F. Ficuciello, L. Villani, and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive human–robot physical interaction," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 850–863, 2015.

[8] S. Jung, T. C. Hsia, and R. G. Bonitz, "Force tracking impedance control of robot manipulators under unknown environment," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 3, pp. 474–483, 2004.

[9] L. Roveda, F. Vicentini, and L. M. Tosatti, "Deformation-tracking impedance control in interaction with uncertain environments," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1992–1997.

[10] L. Roveda, N. Iannacci, F. Vicentini, N. Pedrocchi, F. Braghin, and L. M. Tosatti, "Optimal impedance force-tracking control design with impact formulation for interaction tasks," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 130–136, 2016.

[11] E. Shahriari, A. Kramberger, A. Gams, A. Ude, and S. Haddadin, "Adapting to contacts: Energy tanks and task energy for passivity-based dynamic movement primitives," in *Humanoid Robotics (Humanoids), 2017 IEEE-RAS 17th International Conference on*. IEEE, 2017, pp. 136–142.

[12] A. Kramberger, E. Shahriari, A. Gams, B. Nemec, A. Ude, and S. Haddadin, "Passivity based iterative learning of admittance-coupled dynamic movement primitives for interaction with changing environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6023–6028.

[13] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE control systems magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[14] B. Nemec, M. Simonič, N. Likar, and A. Ude, "Enhancing the performance of adaptive iterative learning control with reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2192–2199.

[15] Y. Li, G. Ganesh, N. Jarrassé, S. Haddadin, A. Albu-Schaeffer, and E. Burdet, "Force, impedance, and trajectory learning for contact tooling and haptic identification," *IEEE Transactions on Robotics*, no. 99, pp. 1–13, 2018.

[16] B. Nemec, T. Petrič, and A. Ude, "Force adaptation with recursive regression iterative learning controller," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2835–2841.

[17] B. Nemec, K. Yasuda, N. Mullennix, N. Likar, and A. Ude, "Learning by demonstration and adaptation of finishing operations using virtual mechanism approach," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7219–7225.

[18] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[19] S. Mirrazavi, M. Khoramshahi, and A. Billard, "A dynamical system approach for catching softly a flying object: Theory and experiment," *IEEE Transactions on Robotics*, vol. 32, 04 2016.

[20] S. S. M. Salehian and A. Billard, "A dynamical system based approach for controlling robotic manipulators during non-contact/contact transitions," *IEEE Robotics and Automation Letters*, 2018.

[21] K. Kronander and A. Billard, "Passive interaction control with dynamical systems," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 106–113, 2016.

[22] M. Khoramshahi, A. Laurens, T. Triquet, and A. Billard, "From human physical interaction to online motion adaptation using parameterized dynamical systems," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1361–1366.

[23] W. Amanhoud, M. Khoramshahi, and A. Billard, "A dynamical system approach to motion and force generation in contact tasks," in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.

[24] K. Kronander, M. Khansari, and A. Billard, "Incremental motion learning with locally modulated dynamical systems," *Robotics and Autonomous Systems*, vol. 70, pp. 52–62, 2015.

[25] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[26] M. D. Buhmann, *Radial basis functions: theory and implementations*. Cambridge university press, 2003, vol. 12.

[27] K. J. Åström and B. Wittenmark, *Adaptive control*. Courier Corporation, 2013.

[28] A. J. van der Schaft and A. Van Der Schaft, *L2-gain and passivity techniques in nonlinear control*. Springer, 2000, vol. 2.

[29] S. Stramigioli, *Energy-Aware Robotics*, 01 2015, pp. 37–50.

[30] C. Secchi, S. Stramigioli, and C. Fantuzzi, "Position drift compensation in port-hamiltonian based telemanipulation," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 4211–4216.

[31] M. Khoramshahi and A. Billard, "A dynamical system approach to task-adaptation in physical human–robot interaction," *Autonomous Robots*, vol. 43, no. 4, pp. 927–946, 2019.

[32] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

[33] W. Martens, Y. Poffet, P. R. Soria, R. Fitch, and S. Sukkarieh, "Geometric priors for gaussian process implicit surfaces," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 373–380, 2016.

View publication stats