Evaluation of Variable Impedance- and Hybrid Force/Motion-Controllers for Learning Force Tracking Skills

Akhil S Anand[1], Martin Hagen Myrestrand[2], Jan Tommy Gravdahl[1]

Abstract— **For robots to perform real-world force interaction tasks with human level dexterity, it is crucial to develop adaptable and compliant force controllers. Learning techniques, especially reinforcement learning, provide a platform to develop adaptable controllers for complex robotic tasks. This paper presents an evaluation of two prominent force control methods, variable impedance control and hybrid force-motion control in a robot learning framework. The controllers are evaluated on a Franka Emika Panda robotic manipulator for a robotic interaction task demanding force and motion tracking using a model-based reinforcement learning algorithm, PILCO. Utilizing the learning framework to find the optimal controller parameters has significantly improved the performance of the controllers. The implementation of the controllers integrated with the robot learning framework is available on `https://github.com/martihmy/Compliant_control`.**

I. INTRODUCTION

In the field of robotics, physical interaction between robot and objects in its environment is a key aspect in solving real-world tasks. Robotic interaction control for robotic manipulators focuses on controlling the dynamic interaction between a manipulator and its environment. Combined with methods of motion control, manipulators can be made capable of following desired trajectories while ensuring a compliant behavior with respect to external forces, providing safe and stable interaction control [1]. Various robotic applications such as industrial assembly, healthcare robotics, human-robot interaction etc. are fundamental robotic force interaction tasks [2]. One key aspect for robots to safely handle such complex real-world applications is to have compliant interaction skills, and therefore advanced force and motion control methodologies are required. Compliant behaviour for robots can be achieved by means of passive mechanical compliance built in to the manipulator, or by active compliance control implemented in the servo control loop, for example, admittance control [3]. Compliant behaviour can also be achieved using direct force control, where Variable Impedance Control (VIC) [4] and Hybrid Force-Motion Control (HFMC) [5] are two prominent approaches.

For robotic systems, learning-based methods offer a platform to design optimal controllers as an alternative to conventional control strategies [2], [6]. While VIC and HFMC are robust interaction control strategies, their performance depends heavily on setting the right parameters according to the environment's compliance properties [7]. Reinforcement Learning (RL) offers a framework to learn flexible controllers [8], which can be used to find optimal parameters for controllers such as VIC and HFMC. In [9], the Natural Actor-Critic algorithm is used in an episodic way to learn the stiffness matrix for VIC. Policy Improvement with Path Integrals (PI^2) algorithm was used in [10], to find the optimal policy parameters for VIC. This approach was later adapted for time-invariant motion representation in [11]. The iterative Linear-Quadratic-Gaussian based learning approach was combined with an operational space HFMC to find the optimal torque control for assembly tasks in [12]. A comparison between joint position, velocity, and torque, and Cartesian pose and VIC controllers were presented in [13]. In [14], the authors used Gaussian Processes (GPs) to model the interaction dynamics and a gradient-based optimization to find the policy.

RL algorithms have proved to be well suited for learning robotic manipulation skills [8] [15]. Model-based RL methods are more relevant for real-world robotic systems as it is highly sample efficient compared to model-free approaches [15], [16]. A model-based RL framework is presented in [17] combining VIC, an ensemble of neural networks to model human–robot interaction dynamics, and an MPC to find the optimal impedance parameters. Among the model-based approaches, the PILCO algorithm is considered to be state-of-the-art for sample efficiency [18]. Even though PILCO is not directly scalable to high dimensional problems and complex dynamical models, it offers a framework to develop and evaluate learning based controllers with minimal robot-environment interactions.

The majority of the learning based methods in robotic interaction control has focused on using VIC to solve specific tasks, with major focus on robotic assembly and human-robot interaction tasks. This paper focuses on implementation and evaluation of two prominent force control approaches (VIC and HFMC) in a model-based learning framework for an interaction task demanding force and motion tracking. The PILCO algorithm is chosen for evaluating the controller considering its high sample efficiency which facilitates learning directly in the experimental set-up in a handful of trails. The force controllers are implemented as Open-AI gym environments to seamlessly integrate with various learning frameworks. The rest of this paper is organised as: Section II presents the necessary background on controller implementation. Section III presents the force control as a learning problem. Section IV presents the evaluation of the controllers in the learning framework and Section V concludes the work.

[1] Akhil S Anand and Jan Tommy Gravdahl are with Dept. of Engineering Cybernetics at Norwegian University of Science and Technology (NTNU), Trondheim, Norway. Contact: `akhil.s.anand@ntnu.no`

[2] Martin Hagen Myrestrand is with Avo Consulting, Bergen, Norway

II. Background

A. Robot manipulator system

1) Task space formulation: For a rigid 6-DOF robotic manipulator, the task space formulation of the robot dynamics is given by,

$$\boldsymbol{\Lambda}(\boldsymbol{q})\dot{\boldsymbol{v}}_e + \boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{v}_e + \boldsymbol{\eta}(\boldsymbol{q}) = \boldsymbol{h}_c - \boldsymbol{h}_e\,, \tag{1}$$

where $\boldsymbol{h}_c$ is the control force, $\boldsymbol{h}_e$ is the external wrench, $\boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{6\times 6}$ is the wrench caused by centrifugal and Coriolis effects, and $\boldsymbol{\eta}(\boldsymbol{q}) \in \mathbb{R}^{6\times 1}$ is the wrench of the gravitational effects. The Cartesian inertia matrix, $\boldsymbol{\Lambda}(\boldsymbol{q}) \in \mathbb{R}^{6\times 6}$ is calculated as,

$$\boldsymbol{\Lambda}(\boldsymbol{q}) = (\boldsymbol{J}\boldsymbol{H}(\boldsymbol{q})^{-1}\boldsymbol{J}^T)^{-1}\,. \tag{2}$$

Where $\boldsymbol{H}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the symmetric and positive-definite joint space inertia matrix and $\boldsymbol{J}$ is the end-effector geometric Jacobian. By additionally knowing the joint space formulation of the centrifugal and Coriolis effects, $\boldsymbol{V}(\boldsymbol{q},\dot{\boldsymbol{q}})$, the corresponding wrench,

$$\boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{J}^{-T}\boldsymbol{V}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{J}^{-1} - \boldsymbol{\Lambda}(\boldsymbol{q})\dot{\boldsymbol{J}}\boldsymbol{J}^{-1}\,. \tag{3}$$

The gravitational wrench is given by,

$$\boldsymbol{\eta}(\boldsymbol{q}) = \boldsymbol{J}^{-T}\boldsymbol{g}(\boldsymbol{q})\,, \tag{4}$$

where $\boldsymbol{g}(\boldsymbol{q})$ is the joint space quantity.

B. Force interaction control for robotic manipulators

In this work we implement two different active interaction control strategies: (i) Variable Impedance Control and (ii) Hybrid Force-Motion Control. Σ represents the robot base frame and Σ_e represents the end-effector frame.

1) Variable Impedance Control: VIC is designed to achieve force regulation by adjusting the system impedance [19], by adapting the inertia, damping and stiffness components. In the presence of a force and torque sensor measuring $\boldsymbol{h}_e$, impedance control can be implemented by enabling inertia shaping [7]. Casting the control law

$$\boldsymbol{h}_c = \boldsymbol{\Lambda}(\boldsymbol{q})\boldsymbol{\alpha} + \boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{\eta}(\boldsymbol{q}) + \boldsymbol{h}_e\,, \tag{5}$$

into the dynamic model in (1) results in, $\dot{\boldsymbol{v}}_e = \boldsymbol{\alpha}$, $\boldsymbol{\alpha}$ being the control input denoting acceleration w.r.t the base frame. Identifying $\dot{\boldsymbol{v}}_e = \bar{\boldsymbol{R}}_e^T\dot{\boldsymbol{v}}_e^e + \dot{\bar{\boldsymbol{R}}}_e^T\boldsymbol{v}_e^e$ with $\bar{\boldsymbol{R}}_e = \text{diag}\ (\boldsymbol{R}_e, \boldsymbol{R}_e)$ and choosing $\boldsymbol{\alpha} = \bar{\boldsymbol{R}}_e^T\boldsymbol{\alpha}^e + \dot{\bar{\boldsymbol{R}}}_e^T\boldsymbol{v}_e^e$, leads to $\dot{\boldsymbol{v}}_e^e = \boldsymbol{\alpha}^e$, where $\boldsymbol{R}_e$ is the rotation matrix w.r.t to the end-effector frame. The control input $\boldsymbol{\alpha}^e$ is the acceleration relative to the end-effector frame. By setting

$$\boldsymbol{\alpha}^e = \dot{\boldsymbol{v}}_d^e + \boldsymbol{K}_M^{-1}(\boldsymbol{K}_D\Delta\boldsymbol{v}_{de}^e + \boldsymbol{h}_\Delta^e - \boldsymbol{h}_e^e)\,, \tag{6}$$

the closed loop expression is found to be

$$\boldsymbol{K}_M\Delta\dot{\boldsymbol{v}}_{de}^e + \boldsymbol{K}_D\Delta\boldsymbol{v}_{de}^e + \boldsymbol{h}_\Delta^e = \boldsymbol{h}_e^e\,. \tag{7}$$

$\boldsymbol{K}_M$ and $\boldsymbol{K}_D$ are symmetric positive-definite matrices, $\Delta\dot{\boldsymbol{v}}_{de}^e$ and $\Delta\boldsymbol{v}_{de}^e$ are the error in acceleration and velocity, and $\boldsymbol{h}_\Delta^e$ is the elastic wrench which is equivalent to a pure moment, all relative to Σ_e. The gain matrices $\boldsymbol{K}_M$, $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are adjustable parameters. With no external wrench acting on the manipulator, under this control scheme the end-effector frame Σ_e asymptotically follows the desired frame Σ_d. In the presence of external forces, the compliant behavior of the end-effector is described by (7), limiting the contact wrench at the expense of a finite displacement in position and orientation.

Consider the position and force errors defined by $\boldsymbol{E}_1 = \boldsymbol{X} - \boldsymbol{X}_d$ and $\boldsymbol{E}_f = \boldsymbol{F}_{ext} - \boldsymbol{F}_d$ respectively, where $\boldsymbol{F}_{ext}$ is equivalent to h_e . The $\boldsymbol{K}_M$, $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are adjusted using the adaptive law proposed in [19],

$$\dot{\boldsymbol{\beta}} = -(\boldsymbol{E}_f^T\boldsymbol{P}\boldsymbol{K}_v^{-1}\boldsymbol{\xi}\boldsymbol{\Gamma}^{-1})^T = -\boldsymbol{\Gamma}^{-1}\boldsymbol{\xi}^T\boldsymbol{K}_v^{-1}\boldsymbol{P}\boldsymbol{E}_f\,. \tag{8}$$

where $\boldsymbol{K}_v \in \mathbb{R}^{n\times n}$ is the gain matrix,

$$\boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{E}_1, \dot{\boldsymbol{E}}_1, \ddot{\boldsymbol{E}}_1)\,, \tag{9}$$

is a $n \times 3n$ matrix and

$$\boldsymbol{\beta} = \boldsymbol{\beta}(\Delta\boldsymbol{K}_M, \Delta\boldsymbol{K}_D, \Delta\boldsymbol{K}_P) \tag{10}$$

is a $3n \times 1$ vector. The corresponding updates are,

$$\begin{aligned} \boldsymbol{K}_M &\to \boldsymbol{K}_M + \Delta\boldsymbol{K}_M\,,\\ \boldsymbol{K}_D &\to \boldsymbol{K}_D + \Delta\boldsymbol{K}_D\,,\\ \boldsymbol{K}_P &\to \boldsymbol{K}_P + \Delta\boldsymbol{K}_P\,. \end{aligned} \tag{11}$$

If the desired contact force $\boldsymbol{F}_d$ is large and the position error $\boldsymbol{E}_1$ is small, the adaptive law will adjust $\boldsymbol{K}_M$, $\boldsymbol{K}_D$ and $\boldsymbol{K}_P$ until $\boldsymbol{F}^* \to \boldsymbol{F}_d$, potentially causing instability issues. Hence, upper bounds should be set for $\boldsymbol{K}_M$, $\boldsymbol{K}_D$ and $\boldsymbol{K}_P$, avoiding instability at the expense of force tracking ability [19].

2) HFMC: First proposed in [5], HFMC aims to achieve both motion and force control by dividing the task into two separate, decoupled sub-problems [7]. By specifying which sub-spaces should be controlled by a motion- and force controller respectively, the hybrid control intend to simultaneously solve the two separate control tasks. The selection matrices $\boldsymbol{S}_v$ and $\boldsymbol{S}_f$ are used to specify these subspaces. In the case of performing force control along the z-axis, and motion control in the remaining five dimensions, given by,

$$\begin{aligned} \boldsymbol{S}_v &= diag\begin{pmatrix}1 & 1 & 0 & 1 & 1 & 1\end{pmatrix}\,,\\ \boldsymbol{S}_f &= \begin{pmatrix}0 & 0 & 1 & 0 & 0 & 0\end{pmatrix}^T\,. \end{aligned} \tag{12}$$

When dealing with a compliant environment, the end-effector twist caused by environmental deformation in the presence of a wrench is given by [7],

$$\boldsymbol{v}_e = \boldsymbol{S}_v\boldsymbol{v} + (\boldsymbol{I} - \boldsymbol{P}_v)\boldsymbol{C}\boldsymbol{S}_f\dot{\boldsymbol{\lambda}}\,, \tag{13}$$

where $\boldsymbol{C} = \boldsymbol{K}^{-1} \in \mathbb{R}^{6\times 6}$ represents the compliance matrix between the end-effector and the environment and $\boldsymbol{\lambda}$ is the force multiplier. $\boldsymbol{P}_v$ is a projection matrix that filters out all the end-effector twists that are not in the range space of $\boldsymbol{S}_v$. $\boldsymbol{I} - \boldsymbol{P}_v$ has the opposite effect of filtering out the twists that are in the range space of $\boldsymbol{S}_v$. $\boldsymbol{P}_v$ is calculated as $\boldsymbol{P}_v = \boldsymbol{S}_v\boldsymbol{S}_v^\dagger$, where $\boldsymbol{S}_v^\dagger$ is a suitable weighted pseudoinverse of $\boldsymbol{S}_v$,

$$\boldsymbol{S}_v^\dagger = (\boldsymbol{S}_v^T\boldsymbol{W}\boldsymbol{S}_v)^{-1}\boldsymbol{S}_v^T\boldsymbol{W}\,. \tag{14}$$

Setting $\boldsymbol{W}$ equal to the inertia matrix $\boldsymbol{H} \in \mathbb{R}^{6\times 6}$ corresponds to defining a norm in the space of twists based on the kinetic energy [7]. Assuming, $\boldsymbol{S}_v$ and compliance,

$$\boldsymbol{C}' = (\boldsymbol{I} - \boldsymbol{P}_v)\boldsymbol{C} \tag{15}$$

to be a constant, (13) leads to the following decomposition of acceleration

$$\dot{\boldsymbol{v}}_e = \boldsymbol{S}_v \dot{\boldsymbol{v}} + \boldsymbol{C}'\boldsymbol{S}_f \ddot{\boldsymbol{\lambda}} \,. \tag{16}$$

Casting the control law (5) into the dynamic model (1) results in, $\dot{\boldsymbol{v}}_e = \boldsymbol{\alpha}$, $\boldsymbol{\alpha}$ is the control input denoting the acceleration with respect to Σ. By choosing,

$$\boldsymbol{\alpha} = \boldsymbol{S}_v \boldsymbol{\alpha}_v + \boldsymbol{C}'\boldsymbol{S}_f \boldsymbol{f}_\lambda \,, \tag{17}$$

allows decoupling of the respective controllers, $\boldsymbol{\alpha}_v$ relating to motion control and $\boldsymbol{f}_\lambda$ to force control. By choosing

$$\boldsymbol{\alpha}_v = \ddot{\boldsymbol{r}}_d(t) + \boldsymbol{K}_{Dr}[\dot{\boldsymbol{r}}_d(t) - \boldsymbol{v}(t)] + \boldsymbol{K}_{Pr}[\boldsymbol{r}_d - \boldsymbol{r}(t)] \,, \tag{18}$$

guarantees asymptotic tracking of desired velocity $\boldsymbol{v}_d$ and acceleration $\dot{\boldsymbol{v}}_d$ with exponential convergence [7]. Choosing,

$$\boldsymbol{f}_\lambda = \ddot{\boldsymbol{\lambda}}_d(t) + \boldsymbol{K}_{D\lambda}[\dot{\boldsymbol{\lambda}}_d - \dot{\boldsymbol{\lambda}}(t)] + \boldsymbol{K}_{P\lambda}[\boldsymbol{\lambda}_d(t) - \boldsymbol{\lambda}(t)] \,, \tag{19}$$

guarantees asymptotic tracking of a desired force trajectory $(\ddot{\boldsymbol{\lambda}}_d(t), \dot{\boldsymbol{\lambda}}_d(t), \boldsymbol{\lambda}_d(t))$, with exponential convergence [7]. $\boldsymbol{K}_{D\lambda}$ and $\boldsymbol{K}_{P\lambda}$ are positive-definite matrices. $\dot{\boldsymbol{\lambda}}$ (19) can be computed from the force measurements $\boldsymbol{h}_e$ as,

$$\dot{\boldsymbol{\lambda}} = \boldsymbol{S}_f^{\dagger} \dot{\boldsymbol{h}}_e \,. \tag{20}$$

where $\boldsymbol{S}_f^{\dagger}$ is the pseudoinverse of $\boldsymbol{S}_f$, computed using, $\boldsymbol{W} = \boldsymbol{C}$ in (14). Due to the noisy force measurements, the estimate

$$\dot{\boldsymbol{\lambda}} = \boldsymbol{S}_f^{\dagger} \boldsymbol{K}' \boldsymbol{J}(\boldsymbol{q}) \dot{\boldsymbol{q}} \,, \tag{21}$$

is often preferred, where $\boldsymbol{K}' = \boldsymbol{P}_f \boldsymbol{K}$ and $\boldsymbol{P}_f = \boldsymbol{S}_f \boldsymbol{S}_f^{\dagger}$.

C. PILCO

PILCO [18] is a data-efficient model-based Policy Search method considered as state-of-the-art in terms of sample efficiency in model-based RL. It is formulated to reduce model bias, one of the key problems of model-based reinforcement learning. This is achieved by learning a probabilistic dynamics model and explicitly incorporating model uncertainty into long-term planning. This way PILCO can cope with a small amount of data, facilitating learning in a handful of trials. Policy evaluation is performed using approximate inference, and policy improvement by computing policy gradients analytically. PILCO considers a dynamic system on the form, $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$ with unknown transition dynamics f, and continuous-valued states $\boldsymbol{x} \in \mathbb{R}^D$ and control-input $\boldsymbol{u} \in \mathbb{R}^F$, where D and F are the dimensions of the state and input space respectively. The objective is to find a policy π that minimizes the expected return

$$J^{\pi}(\theta) = \sum_{t=0}^{T} \mathrm{E}_{\mathrm{x}_t}\left[c\left(\mathbf{x}_t\right)\right], \quad \mathrm{x}_0 \sim \mathcal{N}\left(\mu_0, \boldsymbol{\Sigma}_0\right), \tag{22}$$

over the next T time steps, where $c(\boldsymbol{x}_t)$ is the cost associated with the state $\boldsymbol{x}$ at time t.

III. Learning force tracking

A. Learning Framework

A robot learning framework which can be easily integrated with different RL algorithms is developed in Python. The framework has three components, (i) set of force controllers, (ii) RL algorithms and (iii) robot simulator with the three components interacting with each other. The force controllers (VIC and HFMC) are implemented as OpenAI Gym environments [20] for easy integration with various RL algorithms. A Gazebo-based [21] simulator is setup with a Franka Emika Panda robotic manipulator using the franka simulator framework [22]. For performing the learning trials in simulations and experiments, the framework is integrated with the robotic manipulator using the Franka ROS Interface framework [22]. In both simulation and experimental setups, a python interface is used to fetch the system states and command the control actions.

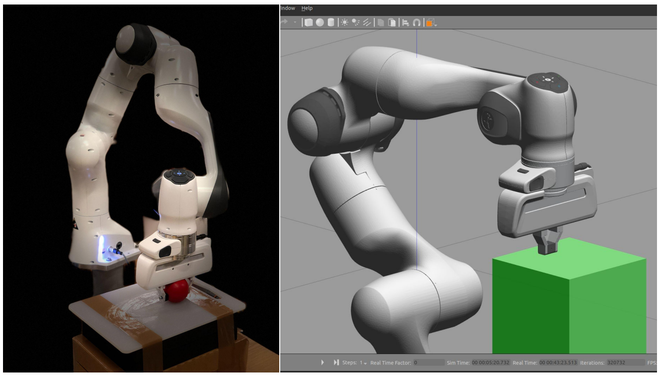

Fig. 1. The setup for testing the force controllers. The experimental set-up is shown on the left and the Gazebo simulator set-up on the right

B. Force Controller Implementation

Accurate modelling of contact interaction behaviour is important in achieving precise force tracking using model-based learning approaches. For modelling the contact transition dynamics, the state vector $\boldsymbol{x}$ was chosen as $\begin{pmatrix} F & p_z & v_z \end{pmatrix}$, where F is the contact force, p_z is the z-position and v_z is the z-velocity all defined in Σ. The action spaces for learning are decided based on the controller, but both the controllers have one action each related to stiffness and damping, in the force tracking direction. Consider an action space, $\boldsymbol{u} \in \mathbb{R}^2$, the training inputs of the model are $(\boldsymbol{x}_t, \boldsymbol{u}_t) \in \mathbb{R}^5$ and the output targets are given by, $\boldsymbol{\Delta}_t = \boldsymbol{x}_{t+1} - \boldsymbol{x} + \boldsymbol{\epsilon} \in \mathbb{R}^3$. Where, $\varepsilon \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}_{\varepsilon}\right), \boldsymbol{\Sigma}_{\varepsilon} = \operatorname{diag}\left([\sigma_{\varepsilon_1}, \ldots, \sigma_{\varepsilon_D}]\right)$. Radial basis functions (RBF) are used to approximate the policy in the simulations and a linear policy is used in the experiments. An exponential cost function is specified based on the desired behaviour of the system with a constant, σ_c deciding the shape of the cost function.

$$c_t = 1 - \exp\left(-\left\|\mathbf{F}_t - \mathbf{F}_d\right\|^2 / \sigma_c^2\right) \in [0, 1] \,. \tag{23}$$

The learning algorithm according to [18] is presented in Algorithm 1. An additional option for using two different

GP contact models is implemented to separately model the transition dynamics for the contact establishment and motion phase.

Algorithm 1 PILCO

```
init: Sample control parameters $\theta \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $\pi(\boldsymbol{\theta})$ and dataset $\mathcal{D}$.
Apply random control signals and record data into $\mathcal{D}$ .
repeat
    Learn probabilistic (GP) dynamics model, using $\mathcal{D}$ .
    Model-based policy search
    repeat
        Approximate inference for policy evaluation, obtain $\mathrm{J}^{\pi}(\theta)$ .
        Gradient-based policy improvement, obtain $\mathrm{d}J^{\pi}(\theta)/\mathrm{d}\theta$ .
        Update parameters $\theta$ .
    until convergence; return $\theta^{*}$;
    Set $\pi^{*} \leftarrow \pi(\theta^{*})$ .
    Apply $\pi^{*}$ to system and record data into $\mathcal{D}$ .
until task learned;
```

VIC implementation: The VIC controller is implemented by adapting the impedance control law in (5) with a modified version of the adaptive impedance law in (8). For force control in z direction, the adaptive laws in (8) only applies to the z-dimensional properties of $(\boldsymbol{K}_M, \boldsymbol{K}_D, \boldsymbol{K}_P) \in \mathbb{R}^{6\times6}$. However, since an adaptive gain matrix, $\boldsymbol{K}_M$ easily can lead to instability, $\boldsymbol{K}_M$ is chosen to be static. This results in a system with adaptive damping and stiffness in z. $\boldsymbol{K}_v$ and $\boldsymbol{P}$ are chosen to be $\boldsymbol{I} \in \mathbb{R}^{6\times6}$ in the adaptive law (8), the law is reduced to,

$$\dot{\boldsymbol{\beta}}(\Delta\dot{\boldsymbol{K}}_D, \Delta\dot{\boldsymbol{K}}_P) = \\ \begin{pmatrix}0... & ... & \gamma_D^{-1}\dot{E}_z E_{fz} & 0... & \gamma_P^{-1} E_z E_{fz} & 0...\end{pmatrix}^T, \quad (24)$$

where E_z is the error in z-position, E_{fz} is the error in z-force, and γ_D^{-1} and γ_P^{-1} are the rates of adaptability for damping and stiffness in z direction respectively. The change in damping and stiffness matrices due to the adaptive law is thus given by,

$$\begin{aligned}\Delta\dot{\boldsymbol{K}}_D &= diag\begin{pmatrix}0 & 0 & \gamma_D^{-1}\dot{E}_z E_{fz} & 0 & 0 & 0\end{pmatrix}, \\ \Delta\dot{\boldsymbol{K}}_P &= diag\begin{pmatrix}0 & 0 & \gamma_P^{-1} E_z E_{fz} & 0 & 0 & 0\end{pmatrix}.\end{aligned} \quad (25)$$

The parameter space for learning are chosen as γ_P and γ_D.

HFMC implementation: In-order to perform force control in z direction and motion control in the remaining five dimensions the selection matrices, $\boldsymbol{S}_f$ and $\boldsymbol{S}_v$ are chosen as, $\boldsymbol{S}_v = \mathrm{diag}(1\ 1\ 0\ 1\ 1\ 1)$ and $\boldsymbol{S}_f = [0\ 0\ 1\ 0\ 0\ 0]^T$. All parameters of the control law (5) are purely state-dependent except the compliance matrix $\boldsymbol{C} = \boldsymbol{K}^{-1}$, the gains of the motion controller $(\boldsymbol{K}_{Dr}, \boldsymbol{K}_{Pr} \in \mathbb{R}^{5\times5})$, and the gains of the force controller $(K_{D\lambda}, K_{P\lambda} \in \mathbb{R})$. The motion controller gains, $\boldsymbol{K}_{Dr}, \boldsymbol{K}_{Pr}$ are kept unchanged as we are interested in improving the force tracking behaviour. Considering the task with a constant desired tracking force, the compliance matrix, $\boldsymbol{C}$ is chosen to be constant. Therefore, $K_{D\lambda}$ and $K_{P\lambda}$ are chosen for tuning. All matrices are chosen to be diagonal, with values derived from testing in simulation and in experiment.

IV. Controller Evaluation

The VIC and HFMC were tested and evaluated in both simulation and experimental set-ups. The controllers were compared based on their performance with and without using the learning framework in Section III-A. A common force-motion tracking task is set up in simulations and experiments. The task is a sweeping task over a flat surface demanding force tracking in the vertical (z) direction and motion tracking in the remaining 5 DOFs (x, y direction and orientations along x, y and z). The controllers are implemented such that only the force tracking parameters are learned by the PILCO algorithm and the motion tracking parameters are kept unchanged. The desired tracking force is 3N while performing a sweeping movement of 5cm over the surface in x direction. The task can be divided into two phases, where in phase 1 the robot establishes a stable contact with the object by achieving a desired contact force. Phase 2 is the motion phase, where the robot performs a sweeping motion across the surface, tracking desired force and motion trajectories.

The controller is considered to be at steady state when the actual contact force has reached a steady state value range around the desired contact force. The force tracking performance is compared by calculating the Mean Squared Error (MSE) between the target and actual contact forces, denoted by Δ_F. The motion tracking behaviour is not considered for evaluation since they are very similar in all the tests as the motion tracking parameters remained unchanged. The simulation and experimental set-ups are illustrated in Fig. 1. In all figures, F_d, denotes the desired force and F, F^{PILCO}, denotes the actual contact force when not-, and when using PILCO to learn the parameters respectively. x, y and x_d, y_d, represents the actual and desired positions in x and y directions respectively. $\Delta q_x, \Delta q_y, \Delta q_z$ represents the differences in orientations along x, y, z directions in quaternions. K_{Pz}, K_{Dz} and ${K_{Pz}}^{PILCO}$, ${K_{Dz}}^{PILCO}$ represents the stiffness and damping in the force tracking direction z, identified without and with learning framework respectively.

A. Simulations

The simulations are performed for force and motion tracking on a flat compliant surface modelled in Gazebo simulator with a stiffness coefficient of $k_p = 5$, and a damping coefficient of $k_d = 3$. In order to test the robustness of the controllers in simulation, a Gaussian noise $\mathcal{N}\left(\mu = 0, \sigma_\varepsilon^2 = 0.015h_e\right)$, is added to force estimate based on comparing with the real system used in experiments. Fig. 2 and Fig. 3 represents both the contact establishment phase and sweeping movement over the surface as shown in the motion trajectories in Fig. 2.c and 3.c.

VIC: Fig. 2 illustrates the force, motion tracking and the varying damping and stiffness in z. The force tracking error, Δ_F in Fig. 2.a is decreased from 0.26 to 0.18 by introducing learning. Steady state is achieved in 0.35s for the learning-based controller compared to 1.5s for the adaptive controller. The mean value of both F and F^{PILCO} is 2.94N and the variances are 0.009 and 0.007 respectively. Fig. 2.b shows how the stiffness and damping in z direction are adapted for achieving the desired force tracking behaviour. The motion tracking behaviour is shown in Fig. 2.c and 2.d.

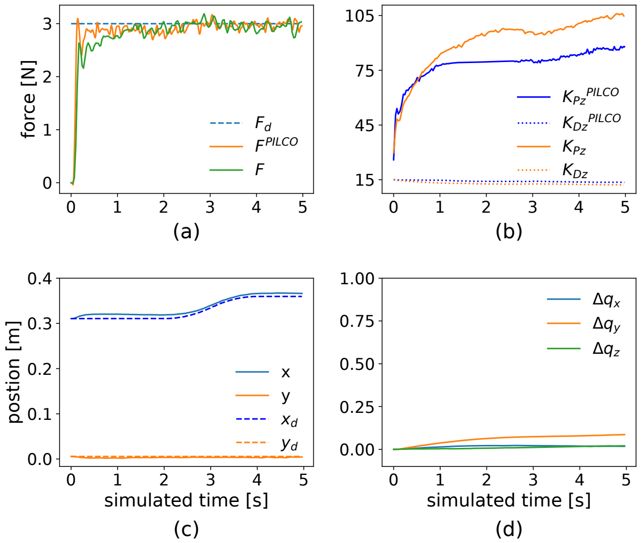

Fig. 2. Simulation results for VIC using PILCO for the set-up in Fig. 1. (a) Force tracking behaviour, (b) stiffness and damping behaviour in z direction, (c) and (d) motion tracking behaviour.

HFMC: The performance of the HFMC in the simulation is shown in Fig. 3, illustrating the force and motion tracking results. Introducing learning has reduced Δ_F from 0.28 to 0.14, but the force tracking behaviour of the learned controller is not smooth as desired. Steady state is achieved in 0.32s for learning-based controller with a mean value of F^{PILCO} is 3.03N and variance of 0.01 compared to 2.77N and 0.03 for F. The motion tracking behaviour is hand-tuned and is comparable with VIC.

B. Experiments

The simulation task is replicated in the experimental setup to perform force-motion tracking on a flat surface. In-order to have a compliant contact between the robot and the rigid surface, the manipulator's end-effector is equipped with a soft ball as shown in Fig. 1. The robot should perform a contact establishment to reach a desired force of 3N and then perform a sweeping motion over the surface for 5cm tracking a desired force. The results of the experiment for both VIC and HFMC in the learning framework is shown in Fig. 4. The introduction of learning in VIC has has reduced Δ_F from 0.13 to 0.04 as shown in Fig. 4.a. During the contact establishment phase, the leaning-based VIC converged to steady state with an overshoot of 0.51N in 1.2s whereas VIC with fixed adaptive law (8) converged to steady state in 3.5s. The learning-based VIC has a steady state variance of 0.002 compared to 0.028 for the VIC with fixed adaptive law. While experimented learning-based HFMC with single GP model for contact establishment and motion phase, the Δ_F was increased to 1.20 compared to 1.17 for the manually tuned HFMC. But this drawback was eliminated by introducing separate contact models for the two phases and thereby significantly reducing the Δ_F to 0.29. By introducing dual contact models for the learning-based HFMC, the Δ_F of the motion phase (phase 2) was decreased from 1.8 to 0.35.

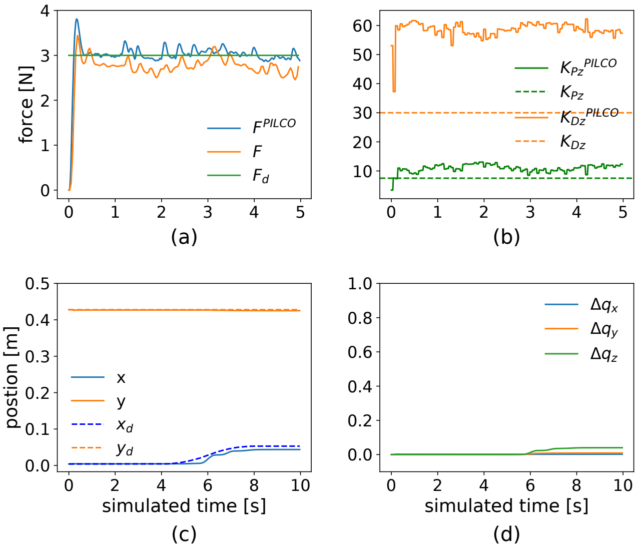

Fig. 3. Simulation results for HFMC using PILCO for the set-up in Fig. 1. (a) Force tracking behaviour, (b) stiffness and damping behaviour in z direction, (c) and (d) motion tracking behaviour.

Fig. 5.a represents how the variance of the learned force GP model changed over the learning iterations in two different experimental trials using learning-based VIC. In Each trial, the experimental task was executed for 14 learning iterations. The trial 1 failed in force tracking with reaching unsafe forces above 5N. Whereas, the successful trial 2 corresponds to the results in Fig. 4.a. Fig. 5.b represents the corresponding change in the cost for the successful trial (trial 2). The decrease in variance after the initial step is minimal during all the successful learning trials conducted with the noisy force data.

C. Discussions

The simulation and experimental evaluation demonstrated the advantage of introducing model-based learning to achieve better force tracking in both VIC and HFMC for the robotic interaction task demanding both force and motion tracking. The simulated and experimental results for VIC, shown in Fig. 2 and 4.a suggests significant improvement in force tracking by introducing model-based learning to adapt the stiffness and damping parameters. The robust adaptive law (8) ensures better force tracking capabilities in VIC even

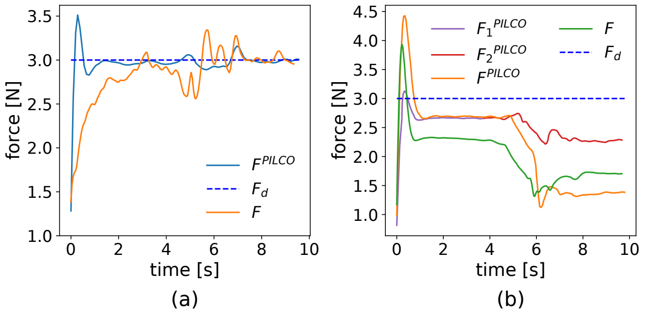

Fig. 4. Experimental results for VIC and HFMC using PILCO for the set-up in Fig. 1. (a) Force tracking behaviour of VIC, (b) force tracking behaviour of HFMC, where $F_1{}^{PILCO}$, $F_2{}^{PILCO}$ represents the contact establishment and motion phase while using separate contact models for both the phases. While, F^{PILCO} represents the force behaviour when using a single model for both the phases as in the case of VIC.

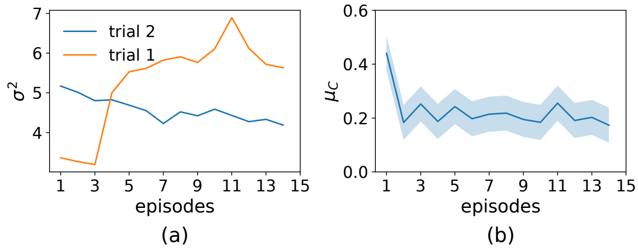

Fig. 5. (a) Variance of the force GP model for two different training trials, (b) cost for the successful trial (trial 2) across the learning iterations.

without introducing learning in simulations. Faster convergence to the steady state is a major impact of learning-based VIC which is consistent in both simulation and experiment. The fast convergence to the desired contact force and less noisy force tracking behaviour is a result of a better impedance strategy optimized by PILCO. Furthermore, the flexibility of the controller allowed it to start off with a high compliance in z direction, avoiding a high initial force overshoot.

The effect of learning in the stiffness and damping parameters are well demonstrated in Fig. 2.b. Similar to VIC, introducing learning has significantly improved the force tracking behaviour of HFMC. The learned stiffness in Fig. 3.b has a noticeable decrease in-order to reduce high impact forces during the contact establishment phase. Introducing learning produced significant improvement the force tracking capabilities as shown in Fig. 3.a except for a higher overshoot in the initial phase. The learned impedance strategy could exploit the force tracking capabilities of HFMC to produce better steady state behaviour both in simulations and experiments. Introducing learning in HFMC had significant improvement in the force tracking error and slightly faster convergence to the desired contact force. The learning-based VIC executed a smooth force tracking behaviour while performing the sweeping motion. Even though the motion tracking ability is not evaluated thoroughly, it was difficult to hand-tune the VIC to achieve accurate motion tracking.

In simulations, a single GP model was used to model both contact establishment phase and motion phase. However, this could not be translated into the experiments because of the frictional effects. The HFMC has no integral action on the force and relies on accurate models to avoid steady state errors. The introduction of separate models for the two phases found to yield better force tracking in the experiments. As in all model-based RL algorithms, the performance of the PILCO algorithm is highly dependent on the accuracy of the dynamics model. The GP model has limited ability in modelling complex dynamics and is not well suited for highly noisy data. Fig. 5.a represents the variance for different GP models across different trials, where difficulty in learning model is translated into the performance as shown using the variances and the cost. The unsuccessful trials failed at reducing the model variances, thereby learning largely uncertain models. This could be improved using more complex models such as ensembles of Neural Networks or Bayesian Neural Networks. There are few promising model-based RL frameworks using such approaches [16], but these methods have a lower sample efficiency compared to PILCO. However, further improvements in model-based RL and the scope of sim2real transfer of task space controllers [13] could realise learning-based force controllers for robotic interaction tasks.

V. CONCLUSIONS

This paper presented the implementation and evaluation of two fundamental approaches in robotic force control, HFMC and a Force-based VIC in the PILCO framework. It was shown that combining a learning-based approach with force controllers has the ability to improve robotic interaction control. For the HFMC, the framework was used to learn direct strategies for its damping- and stiffness parameters. In the VIC, strategies were learned for the parameters of an adaptation law. Both controllers showed significant improvement in force tracking ability by introducing model-based learning. While introducing learning lead to faster convergence to the desired force in VIC, it led to a significant improvement in the force tracking error in HFMC. The results showed that having highly accurate contact dynamics models are key to have accurate force tracking. GP model does not offer the flexibility required to model the complex robot-object contact dynamics. Hence, in future work, we focus on modelling the interaction dynamics using Ensemble of probabilistic neural network and Bayesian Neural Network models in the learning framework with more model-based RL algorithms like PETS [23]. One important aspect of the future work is to improve the sample efficiency while utilizing such complex models.

ACKNOWLEDGMENT

This work was part of the project “Dynamic Robot Interaction and Motion Compensation” funded by the Research Council of Norway under contract number 270941.

References

[1] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[2] F. J. Abu-Dakka and M. Saveriano, "Variable impedance control and learning—a review," *Frontiers in Robotics and AI*, vol. 7, 2020.

[3] M. Schumacher, J. Wojtusch, P. Beckerle, and O. von Stryk, "An introductory review of active compliant control," *Robotics and Autonomous Systems*, vol. 119, pp. 185–200, 2019.

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

[5] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," 1981.

[6] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms." *J. Mach. Learn. Res.*, vol. 22, pp. 30–1, 2021.

[7] L. Villani and J. De Schutter, "Force control," in *Springer handbook of robotics*. Springer, 2016, pp. 195–220.

[8] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[9] B. Kim, J. Park, S. Park, and S. Kang, "Impedance learning for robotic contact tasks using natural actor-critic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 2, pp. 433–443, 2009.

[10] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 820–833, 2011.

[11] J. Rey, K. Kronander, F. Farshidian, J. Buchli, and A. Billard, "Learning motions from demonstrations and rewards with time-invariant dynamical systems based policies," *Autonomous Robots*, vol. 42, no. 1, pp. 45–64, 2018.

[12] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, A. M. Agogino, A. Tamar, and P. Abbeel, "Reinforcement learning on variable impedance controller for high-precision robotic assembly," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3080–3087.

[13] R. Martín-Martín, M. A. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg, "Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1010–1017.

[14] C. Li, Z. Zhang, G. Xia, X. Xie, and Q. Zhu, "Efficient force control learning system for industrial robots based on variable impedance control," *Sensors*, vol. 18, no. 8, p. 2539, 2018.

[15] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 153–173, 2017.

[16] T. Wang, X. Bao, I. Clavera, J. Hoang, Y. Wen, E. Langlois, S. Zhang, G. Zhang, P. Abbeel, and J. Ba, "Benchmarking model-based reinforcement learning," *arXiv preprint arXiv:1907.02057*, 2019.

[17] L. Roveda, J. Maskani, P. Franceschi, A. Abdi, F. Braghin, L. M. Tosatti, and N. Pedrocchi, "Model-based reinforcement learning variable impedance control for human-robot collaboration," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 2, pp. 417–433, 2020.

[18] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*. Citeseer, 2011, pp. 465–472.

[19] H.-P. Huang and S.-S. Chen, "Compliant motion control of robots by using variable impedance," *The International Journal of Advanced Manufacturing Technology*, vol. 7, no. 6, pp. 322–332, 1992.

[20] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[21] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[22] S. Sidhik, "Franka ROS Interface: A ROS/Python API for controlling and managing the Franka Emika Panda robot (real and simulated)." Dec. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4320612

[23] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *arXiv preprint arXiv:1805.12114*, 2018.