2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

A Passivity-based Approach on Relocating High-Frequency Robot Controller to the Edge Cloud

Xiao Chen†, Hamid Sadeghian[1], Lingyun Chen[2], Mario Tröbinger, Abadalla Swirkir[23], Abdeldjallil Naceri and Sami Haddadin[2]

***Abstract*— As robots become more and more intelligent, the complexity of the algorithms behind them is increasing. Since these algorithms require high computation power from the onboard robot controller, the weight of the robot and energy consumption increases. A promising solution to tackle this issue is to relocate the expensive computation to the cloud. In this pioneering work, the possibility of relocating a state-of-the-art nonlinear control is investigated. To this end, the Unified Force-Impedance Controller (UFIC) is relocated to a remote location and high frequency feedback loop is established by including the remote controller in the loop. Passivity analysis is used to ensure the stability of the whole system, comprising the robot in interaction with the environment, the communication channel, as well as the remote controller. The instability associated with the communication channel is resolved by Time Domain Passivity Approach (TDPA). The performance of the proposed framework is experimentally evaluated on a robot arm in interaction with the environment. The results illustrate the stability of the system to a time-varying delay of up to $50 \pm 10ms$.**

I. Introduction

The application of service robots is growing rapidly. Robots are proposed for new near-to-the-human applications every day. Working in dynamic and cluttered human environment demands fast and reliable sensing and advanced control capabilities. The robot needs to perceive the environment through many sensory data and sophisticated algorithms to plan its motion. This is especially the case for humanoid and mobile platforms, which require a lot of onboard computation on the robot controller, which in turn increases the power consumption. On the other hand, this increases the weight of the robot, reduces the agility and mobility of the system, and making the robot expensive. A promising solution is to offload expensive computations to the edge or cloud.

As the technology of cloud computing develops, Kuffner in [1] introduces the *cloud robotics* terminology for the first time. Since then, the idea of offloading computation to the cloud attracted a lot of attention. Algorithms that process large amount of data can benefit from cloud computing, e.g. collective learning algorithms [2], Convolutional Neural Network (CNN), vision-based applications [3] , and motion planning in robotics [4]. Moreover, fast and more reliable wired or wireless communication protocols support this idea and make it feasible.

All authors are with the Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, Munich, Germany, [1] also with Faculty of Engineering, University of Isfahan, 8174673441 Isfahan, Iran, [2] and the Centre for Tactile Internet with Human-in-the-Loop (CeTI), [3] and with the Department of Electrical and Electronic Engineering, Omar Al-Mukhtar University (OMU), Albaida, Libya. † Corresponding Author (xiaoyu.chen@tum.de)

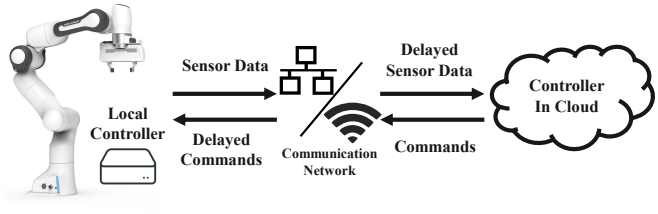

Fig. 1: Framework of relocating robot controller to the cloud.

Relocating high-frequency control to the real-time cloud or edge cloud is a raising hot topic. Edge cloud aims to bring the cloud closer to end-users by moving the computation and storage closer to the network edge. The edge cloud provides a range of benefits, including reduced latency, improved performance, increased reliability, and reduced network congestion. [5], [6] show that under certain criteria, it is even feasible to run high-frequency real-time control loops on the cloud. The first edge-based whole-body control algorithm over a 5G wireless link was presented in [7].

A communication channel is non-passive due to delay [8], making a network-coupled robotic system unstable, especially when interacting with the environment. For a typical telepresence setup, the controller runs on the local computation unit with a direct connection to the robot to ensure system stability and performance [9]. The stability of a tactile robot in teleoperation under commercialized communication network like 5G, LTE, and WiFi has been investigated by the author of this paper in [10]. To ensure the passivity of a network-coupled system, an energy-based Time Domain Passivity Approach (TDPA) is applied to monitor the system energy in real-time using a Passivity Observer (PO) and adapt the controller gain accordingly using a Passivity Controller (PC) [11]. TDPA is a powerful method that can stabilize a system with communication delay and package loss [12]. However, it sacrifices the performance such as position tracking and transparent force feedback. Many algorithms are proposed to reduce the conservatism of TDPA [13] and increase the transparency of a teleoperation system [14].

In this work, we will investigate the possibility of relocating the low-level and high-frequency torque controller into the cloud. The overall system structure is illustrated in Fig. 1. In such a setting, the local controller provides sensory data, and the remote controller receives the data and sends commands back. Note that low-level control algorithm such

as Unified Force-Impedance Controllers (UFIC) [15] rely on a fast and real-time feedback control loop (usually 1kHz). This is different from high-level AI planning services or kinematic controllers, which provides/updates for instance a reference trajectory with a lower sampling rate, and thus hard real-time communication with the onboard robot control is usually not crucial. Here, we aim to address the following main questions:

- Can a tactile robot be controlled by running a low level nonlinear controller on a remote location?
- How is the stability and performance of the system in the presence of delay and packet loss in the communication channel?
- To what extent (entirely or partially), can the controller be offloaded? How is the structure of such a shared control?

Passivity analysis is utilized to establish a framework for such cloud-in-loop control. The general architecture of the system is designed to appear as the interconnection of passive elements, comprising the robot, environment, UFIC controller, and the two-ports communication channel. Because there is no robot dynamics on the cloud, the passivity analysis of UFIC need to be adapted. The quality of communication channel is crucial to the stability and performance of the system, and TDPA is used to ensure its passivity through PO/PC located right before and after this channel. Additionally, the energy tank [15] is used to ensure the passivity of the controller located on the remote side. The UFIC, in fact, only requires low computational power. However, we use it here to demonstrate the feasibility of this framework.

The rest of the paper is organized as follows. Section II describes the modeling of the robot and transmission channel. The main results of the paper are presented in Section III. The performance of the proposed framework is experimentally evaluated in Section IV. Finally, the paper is concluded in Section V.

II. Preliminaries

A. Robot Model

The dynamic of a n Degrees of Freedom (DoF) robot manipulator in the joint space is given by

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_g + \boldsymbol{\tau}_{ext} + \boldsymbol{\tau}_c \quad (1)$$

where $\boldsymbol{q} \in \mathbb{R}^n$ is the joint state, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ and $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ are the inertia and Coriolis/centrifugal matrices, and $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the gravitational torques. The control torque, external torque, and gravity compensation torques are denoted by $\boldsymbol{\tau}_c \in \mathbb{R}^n$, $\boldsymbol{\tau}_{ext} \in \mathbb{R}^n$ and $\boldsymbol{\tau}_g \in \mathbb{R}^n$, respectively.

On the right-hand side of Equation (1), It is assumed that the gravitational torques are compensated by the local controller, which has the full dynamic model of the robot on the local side. Thus, only $\boldsymbol{\tau}_c$ may need high computational power, which is computed in the cloud. Without loss of generality, the robot dynamics is rewritten as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\tau}_{ext} + \boldsymbol{\tau}_c. \quad (2)$$

B. Unified Force Impedance Controller (UFIC)

The classical impedance and force control can be extended and unified into a single framework of unified force impedance control by composition of an impedance control torque $\boldsymbol{\tau}_{imp}$ with a force control torque $\boldsymbol{\tau}_{fc}$ [16],

$$\boldsymbol{\tau}_r = \boldsymbol{\tau}_{imp} + \boldsymbol{\tau}_{fc}. \quad (3)$$

The impedance controller ensures that the robot follows a desired trajectory with the following control law,

$$\begin{aligned}\boldsymbol{\tau}_{imp} =& \boldsymbol{J}^T(\boldsymbol{q})(\boldsymbol{M}_C(\boldsymbol{q})\ddot{\boldsymbol{x}}_{des} + \boldsymbol{C}_C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{x}}_{des} \\ &+ \boldsymbol{K}_x\tilde{\boldsymbol{x}} + \boldsymbol{D}_x\dot{\tilde{\boldsymbol{x}}})\end{aligned} \quad (4)$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x}_{des} - \boldsymbol{x}$ is the tracking error in task space $\boldsymbol{x}$ with the desired trajectory $\boldsymbol{x}_{des}$. $\boldsymbol{J}(q)$ denotes the Jacobian matrix, $\boldsymbol{K}_x$ and $\boldsymbol{D}_x$ are the stiffness and damping matrices, and $\boldsymbol{M}_C(\boldsymbol{q})$ and $\boldsymbol{C}_C(\boldsymbol{q},\dot{\boldsymbol{q}})$ are the robot inertia matrix and Coriolis/centrifugal matrix in Cartesian space.

The force control which regulate the interaction force $\boldsymbol{f}_{ext}$ to following a desired force profile $\boldsymbol{f}_{des}$ is given by

$$\begin{aligned}\boldsymbol{\tau}_{fc} =& \boldsymbol{J}^T(\boldsymbol{q})(\boldsymbol{f}_{des} + \boldsymbol{K}_p(\boldsymbol{f}_{des} - \boldsymbol{f}_{ext}) + \boldsymbol{K}_d(\dot{\boldsymbol{f}}_{des} - \dot{\boldsymbol{f}}_{ext}) \\ &+ \boldsymbol{K}_i\boldsymbol{h}_i(\boldsymbol{f}_{ext},t)),\end{aligned} \quad (5)$$

where $\boldsymbol{h}_i(\boldsymbol{f}_{ext},t) := \int_0^t(\boldsymbol{f}_{des}(\sigma) - \boldsymbol{f}_{ext}(\sigma))d\sigma$. $\boldsymbol{K}_p$, $\boldsymbol{K}_i$ and $\boldsymbol{K}_d$ are the proportional, integral and derivative gains respectively. This controller is intuitive and easy to implement. However, the passivity of the closed-loop system is not preserved during interaction with the environment. An energy tank can be further designed and augmented to the system to ensure the passivity and thus stability of the system [16].

C. Energy Flow Analysis in Two-Port Network

Consider a two-port discrete network system N with ΔT as the sampling period, its energy flow is shown in Fig. 2. The system is passive if and only if at time instance k,

$$\begin{aligned}E(0) + \Delta T\sum_{i=0}^{k}\left(\dot{\boldsymbol{q}}_1^T(i)\boldsymbol{\tau}_1(i) + \dot{\boldsymbol{q}}_2^T(i)\boldsymbol{\tau}_2(i)\right) \geq 0, \\ \forall k \geq 0\end{aligned} \quad (6)$$

holds for velocities $\dot{\boldsymbol{q}}_j$ and torques $\boldsymbol{\tau}_j$, $j \in \{1,2\}$ [11]. The product $\dot{\boldsymbol{q}}_j^T\boldsymbol{\tau}_j$ denotes the power flow at each port and it is denoted positive if energy flows into the network. $E(0)$ is the system initial energy storage and without loss of generality, we assume $E(0) = 0$.

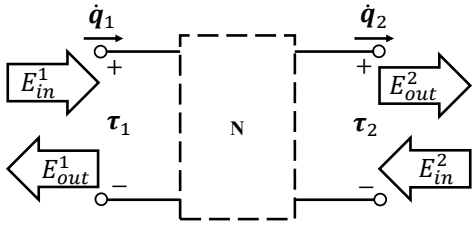

Fig. 2: Energy flow of two-port network. The output energy should be less than the input energy to ensure the network passivity.

In the TDPA, a Passivity Observer (PO) and Passivity Controller (PC) are used to ensure the system time-domain

passivity [12]. The PO observes the system energy in real-time while the PC dissipates the energy generated by the system. For a two-port system, the observed energy, E_{ob}, can be separated as input and output energy at each port, i.e.

$$\begin{aligned} E_{ob}(k) &= \Delta T \sum_{i=0}^{k} \left(\dot{\boldsymbol{q}}_1^T(i)\, \boldsymbol{\tau}_1(i) + \dot{\boldsymbol{q}}_2^T(i)\, \boldsymbol{\tau}_2(i) \right) \\ &= E_{in}^1(k) - E_{out}^1(k) + E_{in}^2(k) - E_{out}^2(k), \end{aligned} \tag{7}$$

where the superscripts represent the port and the subscripts show whether the energy goes in or out of the port. With the power flow defined as $P_1(k) = \dot{\boldsymbol{q}}_1^T(k)\, \boldsymbol{\tau}_1(k)$ and $P_2(k) = \dot{\boldsymbol{q}}_2^T(k)\, \boldsymbol{\tau}_2(k)$ and based on its sign, we can define the input and output powers at each port as

$$\begin{aligned} P_{in}^j(k) &= \begin{cases} P_j(k) & \text{if } P_j(k) > 0 \\ 0 & \text{otherwise,} \end{cases} \\ P_{out}^j(k) &= \begin{cases} -P_j(k) & \text{if } P_j(k) < 0 \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{8}$$

for $j \in \{1, 2\}$. Hence, the input and output energy at each port can be calculated by integrating the power flow as follows,

$$\begin{aligned} E_{in}^j(k) &= \Delta T \sum_{i=1}^{k} P_{in}^j(i) \\ E_{out}^j(k) &= \Delta T \sum_{i=1}^{k} P_{out}^j(i). \end{aligned} \tag{9}$$

III. Proposed Framework

A. General Structure

The general framework structure and data flow between the local controller and the remote controller is illustrated in Fig. 3. The local controller sends the *local* joint velocity $\dot{\boldsymbol{q}}_l$ and the external torque $\boldsymbol{\tau}_{ext}$ to the remote controller in the cloud. On the remote side, the data are received with a forward delay $d_f(t)$, i.e. $\dot{\boldsymbol{q}}_{ld}(t) = \dot{\boldsymbol{q}}_l(t - d_f(t))$ and $\boldsymbol{\tau}_{extd}(t) = \boldsymbol{\tau}_{ext}(t - d_f(t))$, $t \in \mathbb{R}$. After computation on the remote site, the *remote* command torque $\boldsymbol{\tau}_r$ is transferred back to the robot through the communication network with backward delay $d_b(t)$. The robot receives a delayed command torque $\boldsymbol{\tau}_{rd}(t) = \boldsymbol{\tau}_r(t - d_b(t))$.

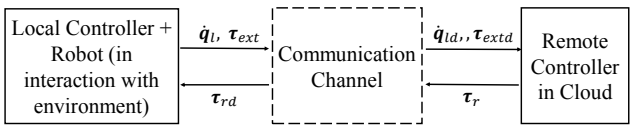

Fig. 3: Data flow between the local controller at the robot side and the remote controller. The communication network introduces time varying delay to the data.

Since in practice, the robot runs in high frequency with constant sampling time, the data transmission actually happens in discrete-time values, $\dot{\boldsymbol{q}}_{ld}(k) = \dot{\boldsymbol{q}}_l(k - d_f(k))$, $\boldsymbol{\tau}_{extd}(k) = \boldsymbol{\tau}_{ext}(k - d_f(k))$, $\boldsymbol{\tau}_{rd}(k) = \boldsymbol{\tau}_r(k - d_b(k))$, $k \in \mathbb{N}$.

Furthermore, the dynamical model of the robot is available in the cloud, in order to acquire the parameters like Cartesian position $\boldsymbol{x}$, inertia matrix $\boldsymbol{M}_C(\boldsymbol{q})$ and Coriolis/centrifugal matrix $\boldsymbol{C}_C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ in Cartesian space. To obtain high accuracy in robot control, the dynamic model of the robot is identified a priori using the approach presented in [17].

B. Passivity Framework for Analysis of the System

The whole system comprising robot/environment, remote controller and the communication channel is arranged in four subsystems in feedback loop interconnection as illustrated in Fig .4. This preserves the passivity of the entire structure while considering the passivity of each subsystem. It is assumed that the robot is in interaction with a passive environment meaning that the environment block is passive w.r.t. input-output pair $(\dot{\boldsymbol{q}}_l, -\boldsymbol{\tau}_{ext})$. Additionally, it is easy to show that the robot dynamics (2) is passive w.r.t. $(\boldsymbol{\tau}_l + \boldsymbol{\tau}_{ext}, \dot{\boldsymbol{q}}_l)$. Therefore, in this section, the passivity of the UFIC with augmented energy tank w.r.t $(-\dot{\boldsymbol{q}}_r, -\boldsymbol{\tau}_r)$ is established. Here, the $\dot{\boldsymbol{q}}_r$ is the modified $\dot{\boldsymbol{q}}_{ld}$, which will be presented in next section. Furthermore, the passivity of Communication Channel combined with time domain PO/PC will also be shown.

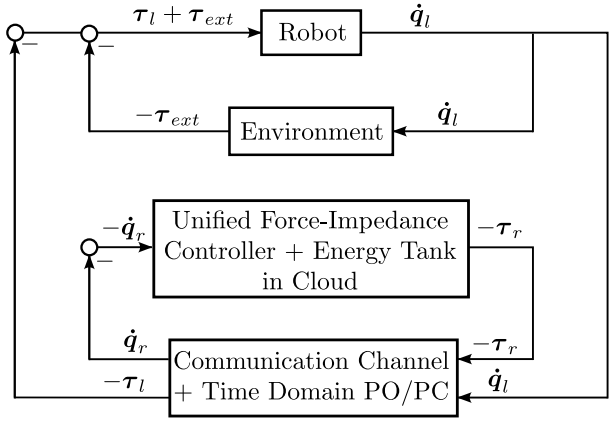

Fig. 4: The block diagram of the system in feedback interconnection of passive elements. The robot in interaction with the environment, as well as the controller, are depicted as one port passive system. The communication channel is considered as two-port passive system.

First, consider the UFIC that is relocated in the cloud, with the storage function defined as $S = \frac{1}{2}\widetilde{\boldsymbol{x}}^T \boldsymbol{K}_x \widetilde{\boldsymbol{x}}$. To show the passivity of the controller, we need to show that its derivative $\dot{S} \leq \dot{\boldsymbol{q}}_r^T \boldsymbol{\tau}_r$. By exploiting (3) and considering the regulation case, i.e. $\dot{\boldsymbol{x}}_{des} = \ddot{\boldsymbol{x}}_{des} = 0$, the following is given

$$\begin{aligned} \dot{S} &= \dot{\boldsymbol{x}}^T \boldsymbol{K}_x \widetilde{\boldsymbol{x}} \\ &= \dot{\boldsymbol{q}}_r^T \boldsymbol{J}^T \boldsymbol{K}_x \widetilde{\boldsymbol{x}} \\ &= \dot{\boldsymbol{q}}_r^T (\boldsymbol{\tau}_r - \boldsymbol{J}^T \boldsymbol{D}_x \dot{\boldsymbol{x}} - \boldsymbol{J}^T \boldsymbol{K}_p (\boldsymbol{f}_{des} - \boldsymbol{f}_{extd}) \\ &\quad - \boldsymbol{J}^T \boldsymbol{K}_d (\dot{\boldsymbol{f}}_{des} - \dot{\boldsymbol{f}}_{extd}) - \boldsymbol{J}^T \boldsymbol{h}_i (\boldsymbol{f}_{extd}, t)) \\ &= (-\dot{\boldsymbol{q}}_r^T)(-\boldsymbol{\tau}_r) - \dot{\boldsymbol{x}}^T \boldsymbol{D}_x \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}^T (\boldsymbol{K}_p (\boldsymbol{f}_{des} - \boldsymbol{f}_{extd}) \\ &\quad + \boldsymbol{K}_d (\dot{\boldsymbol{f}}_{des} - \dot{\boldsymbol{f}}_{extd}) + \boldsymbol{K}_i \boldsymbol{h}_i (\boldsymbol{f}_{extd}, t)). \end{aligned} \tag{10}$$

Where $\dot{\boldsymbol{x}} = \boldsymbol{J}^T \dot{\boldsymbol{q}}_r$. Since the sign of the last term is not clear, the passivity of Unified Force-Impedance Controller

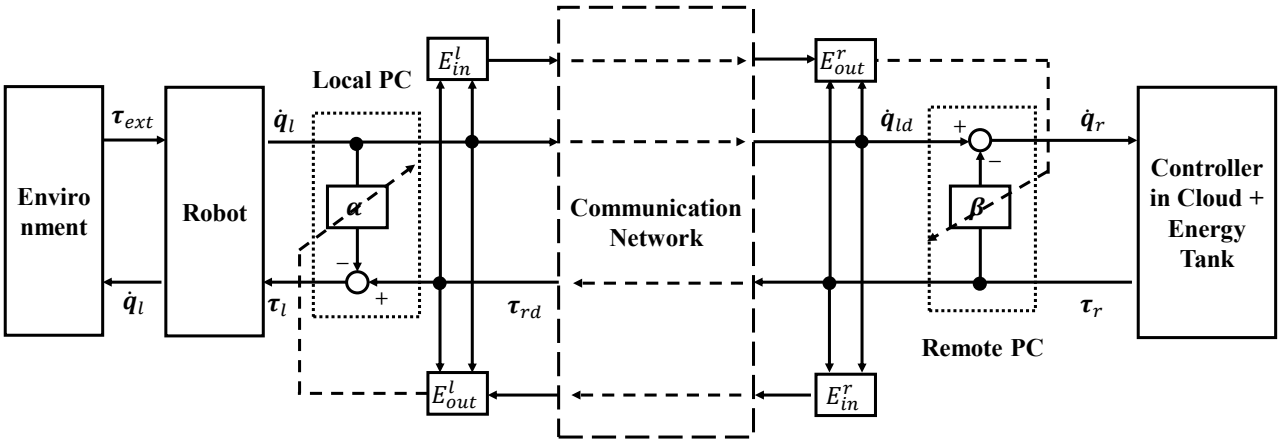

Fig. 5: Diagram of proposed framework with two-port time domain PO/PC

w.r.t $(-\dot{\boldsymbol{q}}_r, -\boldsymbol{\tau}_r)$ is not guaranteed.

To show the passivity of the controller block, an energy tank based method is introduced to modify the control law. Let x_t and $T = \frac{1}{2}x_t^2$ be the tank state and energy, respectively. Also define the tank dynamics as

$$\dot{x}_t = \frac{\zeta}{x_t}(\dot{\boldsymbol{x}}^T \boldsymbol{D}_x \dot{\boldsymbol{x}} + \gamma \dot{\boldsymbol{x}}^T \boldsymbol{K}_d(\dot{\boldsymbol{f}}_{des} - \dot{\boldsymbol{f}}_{extd})) + u_t, \quad (11)$$

where ζ is defined as

$$\zeta = \begin{cases} 1 & \text{if } T \leq T_{max} \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

This indicates that when tank energy exceeds the allowed maximum energy T_{max}, no further energy will be loaded into the tank. The control input of the tank is written as $u_t = -\boldsymbol{\omega}^T \dot{\boldsymbol{x}}$, with

$$\begin{aligned} \boldsymbol{\omega} =& \frac{\eta}{x_t}(\boldsymbol{K}_p(\boldsymbol{f}_{des} - \boldsymbol{f}_{extd}) + (1-\gamma)\boldsymbol{K}_d(\dot{\boldsymbol{f}}_{des} - \dot{\boldsymbol{f}}_{extd}) \\ & - \boldsymbol{K}_i \boldsymbol{h}_i(\boldsymbol{f}_{extd}, t)). \end{aligned} \quad (13)$$

By defining η as

$$\eta = \begin{cases} 1 & \text{if } T \geq T_{min} \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

we can ensure that when the tank energy reaches the allowed minimum energy T_{min}, the force-impedance controller is detached from the energy tank.

The parameter γ is defined to eliminate the negative value and ensure the system passivity,

$$\gamma = \begin{cases} 1 & \text{if } \dot{\boldsymbol{x}}^T \boldsymbol{K}_d(\dot{\boldsymbol{f}}_{des} - \dot{\boldsymbol{f}}_{extd}) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

The original force-impedance controller (3) is now updated with the energy tank as,

$$\begin{aligned} \boldsymbol{\tau}_r =& \boldsymbol{J}^T(\boldsymbol{q}_r)(\boldsymbol{K}_x \widetilde{\boldsymbol{x}} + \boldsymbol{D}_x \dot{\boldsymbol{x}} + \boldsymbol{M}_C(\boldsymbol{q}_r)\ddot{\boldsymbol{x}}_{des} \\ & + \boldsymbol{C}_C(\boldsymbol{q}_r, \dot{\boldsymbol{q}}_r)\dot{\boldsymbol{x}}_{des} + \gamma \boldsymbol{K}_d(\dot{\boldsymbol{f}}_{des} - \dot{\boldsymbol{f}}_{extd}) - \boldsymbol{\omega} x_t) \end{aligned} \quad (16)$$

The following theorem is given.

Theorem 1: The controller given in (16) realizes a passive mapping with respect to input-output pair $(-\dot{\boldsymbol{q}}_r, -\boldsymbol{\tau}_r)$.

Proof: Consider the storage function defined as $S = \frac{1}{2}\widetilde{\boldsymbol{x}}^T \boldsymbol{K}_x \widetilde{\boldsymbol{x}} + \frac{1}{2}x_t^2$. The derivative of this function is derived as follows,

$$\begin{aligned} \dot{S} &= \dot{\boldsymbol{x}}^T \boldsymbol{K}_x \widetilde{\boldsymbol{x}} + x_t \dot{x}_t \\ &= \dot{\boldsymbol{q}}_r^T \boldsymbol{J}^T \boldsymbol{K}_x \widetilde{\boldsymbol{x}} + x_t \dot{x}_t \\ &= \dot{\boldsymbol{q}}_r^T (\boldsymbol{\tau}_r - \boldsymbol{J}^T \boldsymbol{D}_x \dot{\boldsymbol{x}} - \boldsymbol{J}^T \gamma \boldsymbol{K}_d(\dot{\boldsymbol{f}}_{des} - \dot{\boldsymbol{f}}_{extd}) \\ &\quad + \boldsymbol{J}^T \boldsymbol{\omega} x_t) + x_t \dot{x}_t \\ &= \dot{\boldsymbol{q}}_r^T \boldsymbol{\tau}_r - \dot{\boldsymbol{x}}^T (\boldsymbol{D}_x \dot{\boldsymbol{x}} + \gamma \boldsymbol{K}_d(\dot{\boldsymbol{f}}_{des} - \dot{\boldsymbol{f}}_{extd}) - \boldsymbol{\omega} x_t) \\ &\quad + x_t \dot{x}_t \\ &= \dot{\boldsymbol{q}}_r^T \boldsymbol{\tau}_r - \dot{\boldsymbol{x}}^T (\boldsymbol{D}_x \dot{\boldsymbol{x}} + \gamma \boldsymbol{K}_d(\dot{\boldsymbol{f}}_{des} - \dot{\boldsymbol{f}}_{extd}) - \cancel{\boldsymbol{\omega} x_t}) \\ &\quad + \zeta(\dot{\boldsymbol{x}}^T \boldsymbol{D}_x \dot{\boldsymbol{x}} + \gamma \dot{\boldsymbol{x}}^T \boldsymbol{K}_d(\dot{\boldsymbol{f}}_{des} - \dot{\boldsymbol{f}}_{extd})) - \cancel{x_t \boldsymbol{\omega}^T \dot{\boldsymbol{x}}} \\ &= \dot{\boldsymbol{q}}_r^T \boldsymbol{\tau}_r \\ &\quad - (1-\zeta)(\dot{\boldsymbol{x}}^T \boldsymbol{D}_x \dot{\boldsymbol{x}} + \gamma \dot{\boldsymbol{x}}^T \boldsymbol{K}_d(\dot{\boldsymbol{f}}_{des} - \dot{\boldsymbol{f}}_{extd})) \\ &\leq \dot{\boldsymbol{q}}_r^T \boldsymbol{\tau}_r \\ &= (-\dot{\boldsymbol{q}}_r^T)(-\boldsymbol{\tau}_r) \end{aligned} \quad (17)$$

Hence, the block is passive with respect to $(-\dot{\boldsymbol{q}}_r, -\boldsymbol{\tau}_r)$. ■

C. Two-Port TDPA for Communication Channel

The communication channel can be seen as a two-port network. The energy of the communication channel is observed by PO as

$$E_{CC}(k) = E^l_{in}(k) + E^r_{in}(k) - E^l_{out}(k) - E^r_{out}(k) \quad (18)$$

where superscript l indicates the *local side* and r indicates the *remote side*.

In real-world communication network (wired or wireless), problems such as time-varying delay and package loss are inevitable. Therefore, it is impossible to observe the energy at both side at the same time instance. The above issues are the source of instability that is imposed by communication channel and makes the coupled robot system unstable, especially when the robot is interacting with an environment.

Considering the network delay and package loss, and noting that each input energy is monotonous, the sufficient

condition of the system passivity (6) can be written as,

$$\begin{aligned} E^l_{in}(k - d_f(k) - l_f(k)) \geq E^l_{in}(k) \geq E^r_{out}(k) \\ E^r_{in}(k - d_b(k) - l_b(k)) \geq E^r_{in}(k) \geq E^l_{out}(k) \end{aligned} \tag{19}$$

where l_f and l_b are the package loss in forward and backward communication. The PC modifies the torque or velocity to dissipate extra energy. The energy needs to be dissipated at sample time k is calculated as follows,

$$\begin{aligned} E^l_{PC}(k) = E^l_{in}(k - d_f(k) - l_f(k)) - E^r_{out}(k) - E^l_{diss}(k-1) \\ E^l_{PC}(k) = E^r_{in}(k - d_b(k) - l_b(k)) - E^l_{out}(k) - E^r_{diss}(k-1) \end{aligned} \tag{20}$$

The already dissipated energy by the PCs, i.e., $E^l_{diss}(k)$ and $E^r_{diss}(k)$ are given by

$$\begin{aligned} E^l_{diss}(k) = \Delta T \sum_{i=1}^{k-1} \|\boldsymbol{\tau}_r(i)\|^2 \beta(i), \\ E^r_{diss}(k) = \Delta T \sum_{i=1}^{k-1} \|\dot{\boldsymbol{q}}_l(i)\|^2 \alpha(i). \end{aligned} \tag{21}$$

In which the damping element α and β are calculated such that, the condition (19) is satisfied, i.e.,

$$\beta(k) = \begin{cases} \frac{-E^r_{PC}(k)}{\Delta T \|\boldsymbol{\tau}_r(k)\|^2} & \text{if } E^r_{PC}(k) < 0 \\ & \text{and } \boldsymbol{\tau}_r(k) \neq 0 \\ 0 & \text{else} \end{cases} \tag{22}$$

$$\alpha(k) = \begin{cases} \frac{-E^l_{PC}(k)}{\Delta T \|\dot{\boldsymbol{q}}_l(k)\|^2} & \text{if } E^l_{PC}(k) < 0 \\ & \text{and } \dot{\boldsymbol{q}}_l(k) \neq 0 \\ 0 & \text{else} \end{cases} \tag{23}$$

The actual joint velocity $\dot{\boldsymbol{q}}_r(k)$ received by the cloud and the command torque $\boldsymbol{\tau}_l(k)$ received by the robot are calculated based on following PC laws,

$$\begin{aligned} \dot{\boldsymbol{q}}_r(k) = \dot{\boldsymbol{q}}_{ld}(k) - \beta(k)\boldsymbol{\tau}_r(k) \\ \boldsymbol{\tau}_l(k) = \boldsymbol{\tau}_{rd}(k) - \alpha(k)\dot{\boldsymbol{q}}_l(k) \end{aligned} \tag{24}$$

The overall structure of the system including TDPA implementation is depicted in Fig. 5.

The above analysis shows that the Communication Channel with Time Domain PC/PO as illustrated in Fig. 4 is passive with respect to its input and output port. Thus, based on the results from Section III-B, all the subsystems are passive, and thus the entire system is passive.

IV. Experimental Evaluation

A. Experiment Setup

The following experiments serve as the performance evaluation of the proposed control framework. The experiment setup is shown in Fig. 6. Experiments are conducted with a 7-DoF Franka Emika robot [18]. The local and cloud controller run on separate PCs with the same hardware configuration (Intel Core i7-10700 CPU @ 2.90GHz) and operating system (Ubuntu 20.04 LTS with real-time kernel). The PCs are connected via Ethernet and located in the same subnet.

An open source tool called tcgui [19] which utilizes the Linux kernel's network traffic control and shaping features

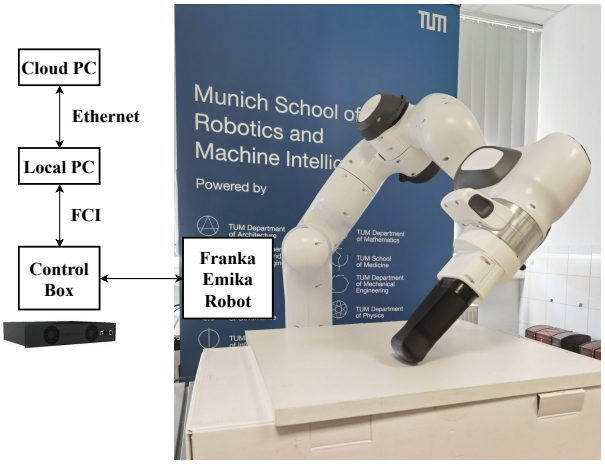

Fig. 6: Experiment setup with Franka Emika Robot; The robot is following a trajectory on the surface plane while regulating the force in the normal direction.

is used to simulate the real-world network condition with variable delay and package loss.

The local PC controls the robot via Franka Control Interface (FCI) at 1 kHz rate. The proposed control framework offloads the UFIC which also runs at 1 kHz rate into the cloud PC. The local PC controls the robot by sending joint torque commands to FCI based on the delayed desired joint torque commands received from the cloud PC. Meanwhile, the robot state acquired from FCI is sent back to the cloud PC. The bi-directional network communication channel between the local and cloud PCs is established using User Datagram Protocol (UDP).

The target of the force-impedance controller is set to follow a circular trajectory with a diameter of $10cm$ and a period of $5s$ in x-y plane and apply a force of $10N$ on the z-axis. The stiffness $\boldsymbol{K}_x$ and damping $\boldsymbol{D}_x$ parameters of the impedance control are set to $diag[400, 400, 400, 1, 1, 1]N/m$ and $diag[10, 10, 10, 0.5, 0.5, 0.5]N \cdot m/s$ respectively. The $\boldsymbol{K}_p$ and $\boldsymbol{K}_i$ for the force controller are set to $0.5\boldsymbol{I}$ and $0.02\boldsymbol{I}$, where $\boldsymbol{I}$ is identity matrix. The setup is tuned at the verge of stability at $50 \pm 10ms$ roundtrip-delay, meanwhile guarantee a good force tracking performance. The network delay parameters from 0 to 50ms with a 10ms interval and 20% variance are set through tcgui. The force measurement is derived from robot joint torque sensors based on the external force estimation method [20].

B. Results

The energy plot in Fig. 7 illustrate the passivity of the communication channel. E_{CC} given in (18) is always greater than zero in all delay scenarios. The accumulated energy also indicates the energy conservatism of the TDPA, as well as the energy tank.

The trajectory of the end-effector and the tracking error are shown in Fig. 8. It is clear that the proposed control framework is capable of controlling the robot to follow the desired trajectory p_{des}. Tracking performance deteriorates with increase of latency. It is obvious when comparing with the error without any delay ep_{std}.

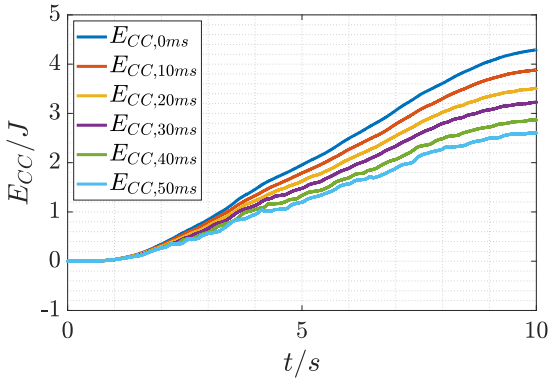

Fig. 7: The observed energy (18) in the communication channel, the positive energy indicates the communication channel is always passive.

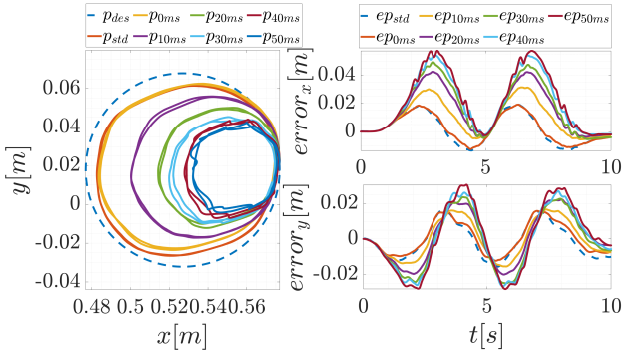

Fig. 8: Position tracking performance. The figure on the left shows the trajectory and figures on the right shows the tracking error in x and y direction. The tracking performance is better with less delay.

The force tracking performance is shown in Fig. 9. With different delay parameters, the proposed framework is able to control the robot to apply desired force profile f_{des}. The force tracking performance also slightly deteriorates with increasing latency (Fig. 10).

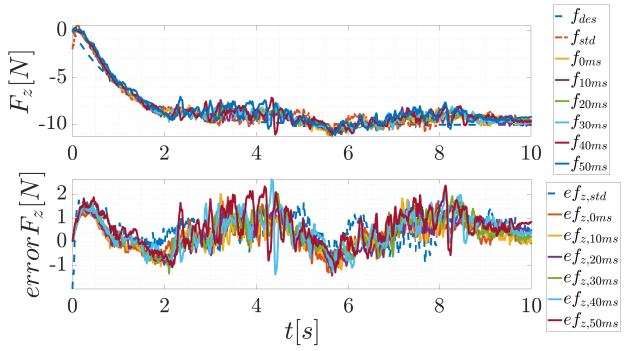

Fig. 9: Force tracking performance. The recorded force is shown in upper plot and the lower plot shows the tracking error.

C. Discussion

Our experiments shows only up to $50 \pm 10ms$ delay, which is at the scale of wired communication at distance of $7000km$ or currently available commercialized 5G network. However, the affordable delay can go higher by tuning the controller parameters.

As the delay increases, the control bandwidth decreases. To increase the total control bandwidth, it is possible to only relocate the computational heavy part of the controller,

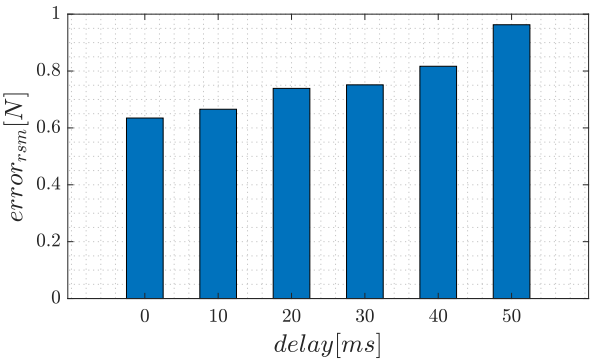

Fig. 10: Root mean square error of force tracking under different delays. The error increases as delay increases.

meanwhile the computational light weight part of that can stay with the local robot. As an example, in this proposed framework, the gravity compensation, which can be computed in short time, is located on the local controller side. In this way, even without receiving any control command, the robot still maintains its basic function.

V. CONCLUSIONS

This article presents a novel framework for relocating computationally intensive controllers to the cloud in order to alleviate the burden on local robots. Our experimental results demonstrate the feasibility of offloading a state-of-the-art nonlinear controller to the cloud for controlling a tactile robot. The proposed passivity-based framework ensures the stability of the system. To maintain passivity in the presence of time-varying communication delays, methods such as the energy tank and TDPA are implemented. Future research will focus on reducing the conservatism of the TDPA to improve tracking performance, as well as exploring how to allocate different components of the controller to the local and remote sides.

ACKNOWLEDGEMENT

This work was funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (BMBF) within the project VERITAS under the Project Number 01IS18073A and also with the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy EXC 2050/1 Project ID 390696704 Cluster of Excellence Centre for Tactile Internet with Human-in-the-Loop (CeTI) of Technische Universitat Dresden. The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany (BMBF) in the programme of "Souverän. Digital. Vernetzt." Joint project 6G-life, project identification number 16KISK002. We gratefully acknowledge the funding of the Lighthouse Initiative Geriatronics by StMWi Bayern (Project X, grant no. IUK-1807-0007// IUK582/001) and LongLeif GaPa gGmbH (Project Y). We also gratefully acknowledge the funding by the European Union's Horizon 2020 research and innovation program as part of the project ReconCycle under grant no. 871352. Please note that S. Haddadin has a potential conflict of interest as shareholder of Franka Emika GmbH.

REFERENCES

[1] J. KUFFNER, "Cloud-enabled humanoid robots," *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on, Nashville TN, United States, Dec.*, 2010.

[2] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[3] Z. Kamranian, H. Sadeghian, A. R. Naghsh Nilchi, and M. Mehrandezh, "Fast, yet robust end-to-end camera pose estimation for robotic applications," *Applied Intelligence*, vol. 51, no. 6, pp. 3581–3599, 2021.

[4] A. Vick, V. Vonásek, R. Pěnička, and J. Krüger, "Robot control as a service—towards cloud-based motion planning and control for industrial robots," in *2015 10th International Workshop on Robot Motion and Control (RoMoCo)*. IEEE, 2015, pp. 33–39.

[5] D. M. Lofaro, A. Asokan, and E. M. Roderick, "Feasibility of cloud enabled humanoid robots: Development of low latency geographically adjacent real-time cloud control," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 519–526.

[6] D. M. Lofaro and A. Asokan, "Low latency bounty hunting and geographically adjacent server configuration for real-time cloud control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5277–5282.

[7] H. Zhu, M. Sharma, K. Pfeiffer, M. Mezzavilla, J. Shen, S. Rangan, and L. Righetti, "Enabling remote whole-body control with 5g edge computing," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3553–3560.

[8] R. Anderson and M. Spong, "Bilateral control of teleoperators with time delay," *IEEE Transactions on Automatic Control*, vol. 34, no. 5, pp. 494–501, 1989.

[9] L. Chen, A. Swikir, and S. Haddadin, "Drawing elon musk: A robot avatar for remote manipulation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4244–4251.

[10] X. Chen, L. Johannsmeier, H. Sadeghian, E. Shahriari, M. Danneberg, A. Nicklas, F. Wu, G. Fettweis, and S. Haddadin, "On the communication channel in bilateral teleoperation: An experimental study for ethernet, wifi, lte and 5g," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, p. in progress.

[11] B. Hannaford and J.-H. Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, 2002.

[12] J.-H. Ryu, J. Artigas, and C. Preusche, "A passive bilateral control scheme for a teleoperator with time-varying communication delay," *Mechatronics*, vol. 20, no. 7, pp. 812–823, 2010, special Issue on Design and Control Methodologies in Telerobotics.

[13] M. Panzirsch, J.-H. Ryu, and M. Ferre, "Reducing the conservatism of the time domain passivity approach through consideration of energy reflection in delayed coupled network systems," *Mechatronics*, vol. 58, pp. 58–69, 2019.

[14] H. Singh, A. Jafari, and J.-H. Ryu, "Enhancing the force transparency of time domain passivity approach: Observer-based gradient controller," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1583–1589.

[15] C. Schindlbeck and S. Haddadin, "Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 440–447.

[16] E. Shahriari, L. Johannsmeier, E. Jensen, and S. Haddadin, "Power flow regulation, adaptation, and learning for intrinsically robust virtual energy tanks," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 211–218, 2019.

[17] A. Jubien, M. Gautier, and A. Janot, "Dynamic identification of the kuka lwr robot using motor torques and joint torque sensors data," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 8391–8396, 2014.

[18] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jaehne, L. Hausperger, and S. Haddadin, "The franka emika robot: A reference platform for robotics research and education," *IEEE Robotics & Automation Magazine*, 2022.

[19] B. Heß, C. Sieber, M. Hofbauer, and TUM LKN team, "tcgui: A lightweight Python-based Web-GUI for Linux traffic control," 7 2016.

[20] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the dlr-iii lightweight manipulator arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 1623–1630.