# Remote Human-to-Machine Distance Emulation through AI-Enhanced Servers for Tactile Internet Applications

**Sourav Mondal, Lihua Ruan, and Elaine Wong**
*Department of Electrical and Electronic Engineering, The University of Melbourne, VIC 3010, Australia.*
*{smondal, ruanl}@student.unimelb.edu.au; ewon@unimelb.edu.au*

**Abstract:** We alleviate the master-slave distance limitation of human-to-machine applications by forecasting and pre-empting haptic feedback transmission. Results show 99% accuracy in detecting touch events and 96% accuracy in forecasting feedback from different slave materials.


## 1. Introduction

The *Tactile Internet* envisions a telecommunication network that supports and empowers human users to immersively control and manipulate both real and virtual objects/machines [1]. Thus, central to the Tactile Internet is human-to-machine (H2M) communications, which in turn comprise a master controller device that sends position and velocity data over the communication network and a slave teleoperator device that responds back with haptic feedback, video, and audio data. The haptic feedback contains both kinesthetic perception and tactile perception data that needs to be transmitted over a secured telecommunication link with ultra-high reliability of guaranteed 99.999% and at very low latency in the order of 1 ms [3].

Fig. 1 illustrates an example network architecture where different combinations of *local and remote H2M master-slave pairs* are connected over fiber-wireless (FiWi) access networks. While communication between local master-slave pairs residing within the same wireless coverage can be supported only by advanced wireless technology such as 5G, communication between remote master-slave pairs need to traverse optical front/back-haul segments [4]. Thus, it becomes challenging to meet the stringent latency requirements of H2M applications without limiting the master-slave distance. This challenge can be overcome by pre-empting the haptic feedback from slave devices and transmit this feedback from an intermediate artificial intelligence (AI)-enhanced H2M server, as shown in Fig. 2. With this approach, the master receives the haptic feedback samples much quicker than the round-trip time of the master-slave pair and thus, remote H2M communications that meet stringent latency requirements can be deployed without being limited by the master-slave distance.

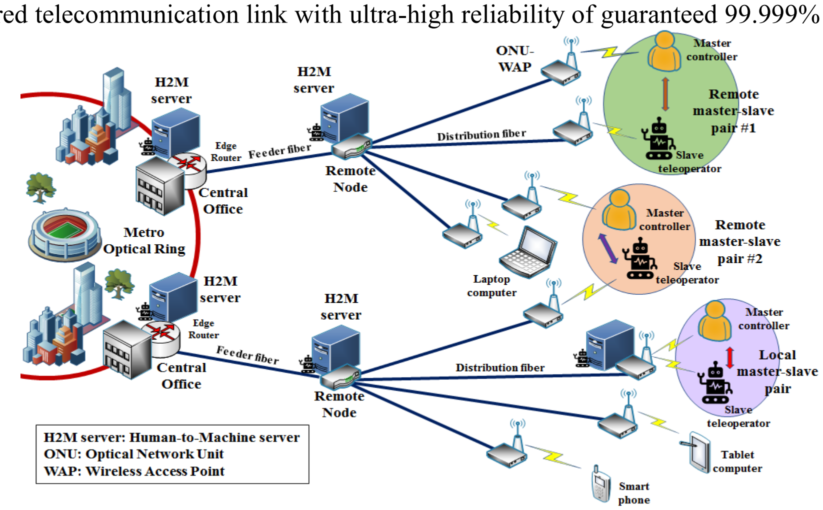


Fig. 1. Local and remote teleoperation master-slave pairs over FiWi networks.

The authors of [3] proposed an edge sample forecast module that uses relies on historical data to forecast haptic feedback to the master. However, user experience can be significantly impacted by dynamic H2M applications such as teleoperation. Thus, in this paper, we propose an *Event-based HAptic SAmple Forecast (EHASAF)* module that exploits a two-stage AI model consisting of an ANN unit followed by a reinforcement learning (RL) unit to forecast haptic feedback to the master. The ANN unit decides when the master controller should start receiving haptic feedback samples and the RL unit ensures that the proper values of the haptic feedback samples are delivered. Thus, the master controller receives proper haptic feedback samples within the expected time ( $D_{QoE}$ ) that defines satisfactory quality of experience (QoE) and the QoE improves significantly. Results show that with

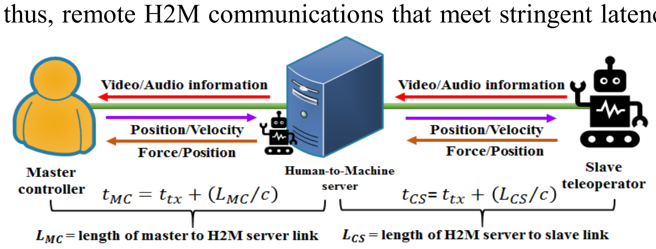


Fig. 2. Logical links among master, H2M server, and slave.

EHASAF, events can be detected with ~99% accuracy and haptic feedback samples can be forecast with ~96% accuracy against 2 different types of slave materials and with ~87% accuracy with 4 different types of slave materials.

## 2. Experimental Setup

To study the contents of the control signals from a master device and the corresponding haptic signals from a slave device, we created an experiment, in which setup is shown in Fig. 3. The master device consists of a pair of virtual reality (VR) gloves and each glove has two orientation sensors on the thumb and wrist with 9 degrees-of-freedom, and five flexible sensors on five fingers for tracking movements and applied forces [5]. The sensor sampling rate is 200 Hz and the control signals are transmitted over a wireless interface to the computer where a VR application of touching a virtual ball (slave device) is run. Quaternion and Euler angles are used to record the thumb, wrist, and finger joints' orientation data [6]. Moreover, two flex sensors per finger record the normalized tension on each finger. So, each instance of control signal from each hand contains a total of (4 × 2) + (5 × 5 × 3) + (5 × 2) = 93 elements.

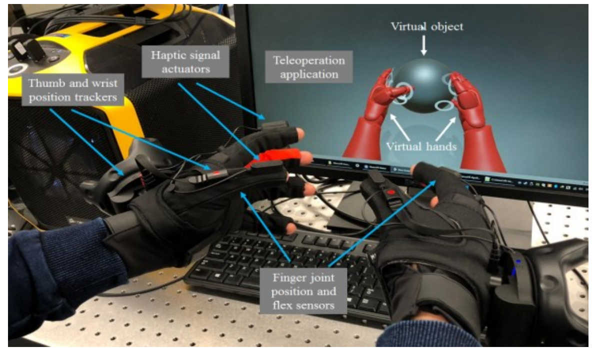


Fig. 3. Experimental setup demonstrating H2M applications.

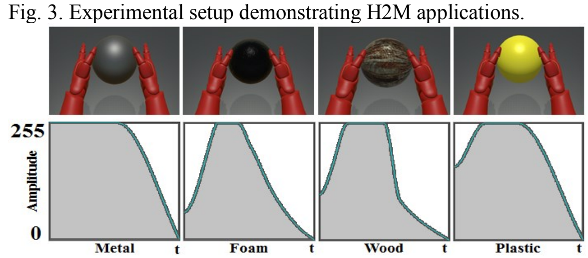


Fig. 4. Haptic feedback for different types of materials.

When any finger touches the virtual ball and depending on the type of material of the ball, e.g., metal, foam, wood, or plastic, the VR application sends different haptic feedback samples, as shown in Fig. 4, to the haptic actuators of the corresponding finger. The haptic feedback samples, with amplitude values that lie within 0 to 255, are transmitted sequentially to the master device over sequential time-slots. The signal latency of the haptic feedback is 10 ms and the maximum force felt is 0.9 gram-force [5].

## 3. Principle of Event-based Haptic Sample Forecasting

The communication among master device, H2M server, and slave device can be summarized by the logical links shown in Fig. 2. The latency between master and H2M server is denoted as $t_{MC}$ and between H2M server and slave is denoted as $t_{CS}$. Hence, the total end-to-end latency between a control signal generation at the master device and reception of the corresponding haptic signal is $D_{MS} = 2(t_{MC} + t_{CS})$. However, if $D_{MS} > D_{QoE}$, then the user experience degrades.

To overcome this issue, we install the proposed EHASAF module in the H2M server that acts as a proxy of the slave teleoperator. This module has an ANN unit corresponding to each of the thumb, index, middle, ring, and baby fingers of both the hands (as shown in Fig. 5) and detect each finger's actions, i.e., whether it is going to touch the virtual object or not. The wrist and thumb coordinates and the rotation and tension of each finger are considered as inputs to the ANN. The ANN corresponding to each finger uses supervised learning and acts as a *binary classifier* [7]. Hence, the outputs of each ANN are *touch* and *no-touch*. If it is a no-touch event, then no immediate haptic feedback is required to be transmitted to the master. However, when any finger touches the object, the EHASAF module starts to generate haptic samples every ($D_{QoE} - t_{MC}$) interval. As such, the master device receives a haptic feedback sample after every $D_{QoE}$ interval, meeting both user experience and network latency constraints.

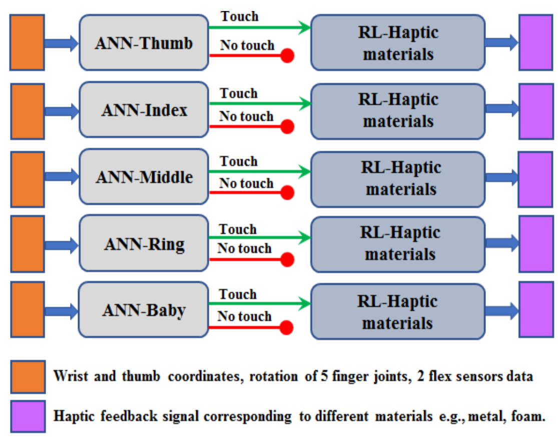


Fig. 5. ANN and RL units for EHASAF.

However, when there are different types of materials involved, it is important to correctly forecast the corresponding haptic feedback samples. Hence, we implement the RL unit that uses the *linear reward-inaction algorithm* for this purpose [8]. When a finger touch is detected by the ANN at $i^{th}$ time-slot, the RL units randomly chooses a material ($m_k^{(i)}$), where $k \in \{1, \ldots, K\}$ and forecasts the first haptic sample. After every ($t_{MC} + 2 \times t_{CS}$) interval, the H2M server receives the actual haptic feedback sample from the slave device and computes the reward

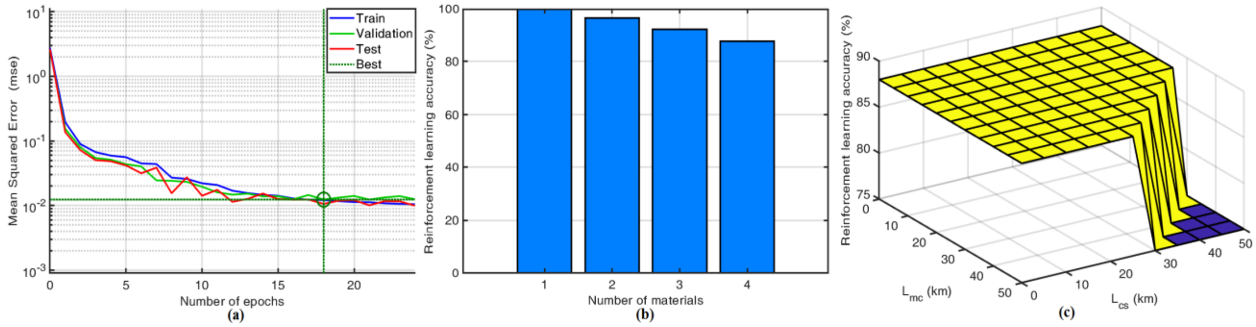


Fig. 6. (a) Accuracy of the binary classification by ANN (~99%), (b) RL accuracy against total number of materials, and (c) RL accuracy with 4 different materials against distance of H2M server from master and slave devices.

($r^{(i)}$), which is the normalized error in haptic sample forecasting. With this reward value, the RL unit updates its probability distribution for choosing a haptic material in the ($i$+1)$^{\text{th}}$ timeslot. When $\left(D_{QoE} - t_{MC}\right) \geq \left(t_{MC} + 2 \times t_{CS}\right)$, then the H2M server receives the haptic feedback sample before generating the next haptic sample and RL unit works at its best; otherwise the RL unit takes more time to detect the actual material and the user experience consequently degrades. A summary of the working principle of the RL unit, if a touch event is detected by ANN at the $i^{\text{th}}$ time-slot, is given as follows:

- The RL unit randomly chooses a material $m_k^{(i)}$ from all possible $K$ materials.
- We denote the haptic feedback sample chosen by the RN unit at $i^{\text{th}}$ time-slot by $h_{cm}^{(i)}$ and the actual haptic sample generated at the slave device by $h_{tm}^{(i)}$. Thus, we define the value of the reward received at the H2M server as, $r^{(i)} = \left(h_{cm}^{(i)} - h_{tm}^{(i)}\right) \big/ 255$, such that $0 \leq r^{(i)} \leq 1$.
- Thus, the probability distribution of the RL unit for choosing a material in ($i$+1)$^{\text{th}}$ timeslot is updated as follows:

$$\boldsymbol{p}_m^{(i+1)} = \boldsymbol{p}_m^{(i)} + \alpha r^{(i)} \left(\boldsymbol{e}_k^{(i)} - \boldsymbol{p}_m^{(i)}\right) \tag{1}$$

  where $0 \leq \alpha \leq 1$ is the *learning rate parameter* and $\boldsymbol{e}_k^{(i)}$ is an *indicator vector* with unity at the $k^{\text{th}}$ index.

## 4. Performance Evaluations

From our experimental setup, firstly we collected 800 instances of both control and haptic data from touching a virtual ball of different materials with random fingers of both hands. The training data for each ANN corresponding to each finger contains (5 × 3) + 2 = 17 columns. We implemented an ANN in MATLAB that contains 2 hidden layers with 10 and 5 nodes, respectively and used the *Levenberg–Marquardt* training method for training because of its faster convergence rate as compared to conventional gradient descent algorithms [9]. Fig. 6(a) shows the accuracy of the binary classification performed by the ANN with the control data from the left thumb. The mean square error decreases gradually with the number of epochs. We used 70% of the data for training, 15% data for validation, and 15% for testing to achieve a prediction accuracy of ~99%.

When the ANN detects a finger touching the virtual ball, EHASAF starts to generate haptic feedback samples. With only one material involved in the testing, the feedback is 100% correct, but if there are multiple materials to choose, then the accuracy of the forecasted haptic feedback samples decreases. Fig. 6(b) shows that the accuracy of the RL unit against the number of materials and it is interesting to note that even with 4 different materials, the RL accuracy is around 87%. Finally, we show the role of master-slave distance on the RL accuracy with $D_{QoE}$ = 1 ms and 4 different materials in Fig. 6(c). We vary the length of master-H2M server and slave-H2M server links from 0 to 50 km The RL unit performs at its optimum until the sum of $L_{MC}$ and $L_{CS}$ is ~80 km. Beyond this aggregated distance, the actual haptic feedback samples from the slave are delayed to the H2M server and the RL algorithm's performance degrades, but the EHASAF module still ensures that master receives feedback samples within $D_{QoE}$.

## 5. References

[1] G. P. Fettweis, *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64-70, March 2014.
[2] K. Antonakoglou *et al.*, *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3034-3059, Fourth quarter 2018
[3] M. Maier *et al.*, *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, no. 4, pp. B10-B25, April 2019.
[4] L. Ruan *et al.*, *IEEE Networking Letters*, vol. 1, no. 3, pp. 128-131, Sept. 2019.
[5] Manus VR gloves. [Online]. https://manus-vr.com/
[6] Quaternions and spatial rotation. [Online]. https://en.wikipedia.org/wiki/Quaternions_and_spatial_rotation
[7] M. A. Nielsen, *Neural Networks and Deep Learning*, Determination Press, 2018.
[8] K. S. Narendra *et al.*, *Learning Automata: An Introduction*, Englewood Cliffs, NJ, Prentice-Hall, 1989.
[9] H. Liu, *2010 Sixth International Conference on Natural Computation*, Yantai, 2010, pp. 456-460.