



Tong, X., Meng, Z., Zhao, G. , Li, L. and Chen, Z. (2021) How to Quantify Packet Importance for Real-Time Control: A Feature-Oriented Perspective. In: 17th IEEE International Conference on Factory Communication Systems (WFCS 2021), 09-11 Jun 2021, ISBN 9781665424790 (doi:[10.1109/WFCS46889.2021.9483612](https://doi.org/10.1109/WFCS46889.2021.9483612))

The material cannot be used for any other purpose without further permission of the publisher and is for private use only.

There may be differences between this version and the published version. You are advised to consult the publisher's version if you wish to cite from it.

<http://eprints.gla.ac.uk/240712/>

Deposited on 04 May 2021

Enlighten – Research publications by members of the University of
Glasgow
<http://eprints.gla.ac.uk>

How to Quantify Packet Importance for Real-Time Control: A Feature-Oriented Perspective

Xin Tong*, Zhen Meng[†], Guodong Zhao[†], Liying Li[‡], and Zhi Chen*

*National Key Lab. on Communications, UESTC, Chengdu, China

[†]James Watt School of Engineering, University of Glasgow, Glasgow, UK

[‡]Mathematics, Physics and Electrical Engineering, Northumbria University, Newcastle, UK

Abstract—Fueled by ubiquitous connectivity, packets are expected to be timely updated to the controller of interest in real-time control systems. Recently, the Age of Information (AoI) becomes a popular metric to quantify the packet importance, which improves the efficiency of communication resource utilization. However, it is analyzed only from the information freshness perspective, losing sight of considering feature data. In light of this, we first establish a feature-oriented teleoperation framework to quantify the packet importance, which is derived from the information bottleneck principle. Under this framework, a packet management method is proposed to increase the average feature quantity of the receiver. Finally, we build a prototype to deploy the proposed method, and the results show superiority in reducing both the communication traffic and the control error.

I. Introduction

The proliferation of sensing, calculating, and communication is revolutionizing the way of how people carry out their work, in particular in cyber-physical systems and industrial internet-of-things [1] [2]. In this scope, future applications including Healthcare 4.0 [3] [4], V2X [5], and remote surgery [6] are emerging, which significantly facilitates the development of economic and flourishes productivity.

In these applications, fueled by ubiquitous connectivity, communication packets with information ranging from environment status to humans' behaviour are expected to be timely delivered to the receiver of interest. For example, considering a typical industrial scenario, where the robotic arm soldering circuit boards controlled by teleoperator shown as Fig. 1, timely update of the time-sensitive information provides accurate operation reference, enabling teleoperators to interact with equipment and perform a delicate task in real-time fashion. Also, teleoperators are relieved from tedious effort duplication by implementing highly mechanized and automatized processes, which improves work efficiently and flexibility.

However, such mission-critical applications require stringent Quality of Service (QoS) of packet transmission, which brings unprecedented research challenges in the communication design. Based on the 5G standards, such applications usually need to enable less than 1ms cycle time and lower than 1 μ s jitters with approximately 50 nodes per transmission in normal work conditions [7]. In

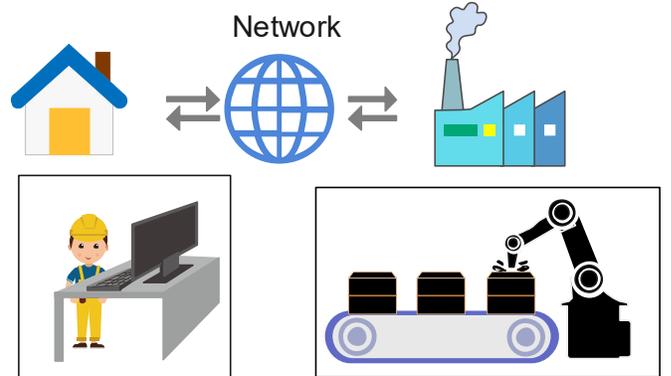


Fig. 1. A typical teleoperation scenario.

order to achieve this target, a large amount of wireless resource overhead is required to be provided, which is not affordable for industrial scenarios [8]. Also, it is clumsy to treat all information as equally important and carry it with the same massive resource overhead, since some packets are relatively not of high priority in certain circumstance. Recall the example shown in Fig. 1, in the whole welding task, we pay more attention to the accuracy of completing the welding rather than the accuracy of the movement process of the manipulator from the initial point to the welding point. Intuitively, a communication package used to complete high-precision tasks is more important, which gives an opportunity to manage communication. Therefore, a more intelligent communication system can be foreseen, where the priority and importance of packets in the communication system are enabled to be perceived.

How to quantify the importance of a packet? Motivated by the requirement of keeping data fresh, a recently popular metric age of information (AoI) has been proposed, which is to quantify the packet importance from the perspective of the increment of information [9]- [15]. However, the relationship between AoI and the incremental information can not be easily characterized since information grows without following any specific pattern in a real environment. To address this, a more in-depth concept, the mutual information (MI) [14], is adopted to mathematically quantifies the amount of the increment of information. Nevertheless, these methods were derived

only from the information perspective, losing sight of considering feature data, which are always generated during the critical stage. e.g., the welding process in Fig.1 in terms of the entire task.

In light of this, an elaborated strategy is required to be designed from a feature-oriented perspective, i.e., taking account of how the receiver exploits the delivered information. Related work has been discussed in control systems, e.g., how AoI affects the actuator (receiver) in wireless control systems was discussed in [12] [13], and the relationship between AoI and the level of stability in event-trigger control systems was discussed in [14] [15]. However, these methods were discussed in specific control models, where the controller and the plant pattern are highly simplified, which is not feasible in real application scenarios, especially in the trends of replacing traditional controllers with the Machine Learning (ML) module. Also, the information carried in the packet is usually assumed as the one-dimensional information, while the information frame in real-word consists of high-dimensional data that includes not only feature but also irrelevant data. These data potentially interfere with the importance evaluation. Additionally, it is difficult to know which part of information is the feature since the ML module is a black-box. Therefore, it is highly challenging to quantify the importance of a packet in a real application system.

In the paper, we firstly propose to utilize the information bottleneck (IB) [16] principle to address the above issues together. In an IB perspective, the black-box can be explained as an encoder and a decoder where the high-dimensional information is going to be compressed to a low-dimensional feature by the encoder. The feature preserves most of the valuable information and discards the useless and irrelevant data. Thus, we propose to use the incremental feature as the metric for quantifying the information importance, i.e., the importance of the packet. After that, we establish a prototype to verify the effectiveness of the proposed metric in a practical application scenario, where sensing, communication with packet management, and robotic control are included. The experiment results show that the packet management based on the proposed metric effectively reduces the communication traffic up to 25% and in the meantime reduces the control error up to 23.5%.

The rest of this paper is organized as follows. In Section II, we summarize the most similar work as this paper to highlight the contribution of this paper. In Section III, we illustrate the teleoperation framework and propose the metric for quantifying the information importance. In Section IV, we establish a prototype to verify our method in a practical application scenario. In Section V, experiment results and the corresponding discussion are provided, which shows that the proposed method improves control performance and reduces network traffic. Finally, Section VI concludes the paper.

II. Related Work and Contribution

Significant research has been conducted to discuss packet importance. The most relevant papers are [11] [12] [13] [17], which utilizes the incremental information to quantify the packet importance, while most of them only from a source perspective. Even though some papers discusses how the information affects control systems, only simple control models are used, such as [12] [13] [15] [14]. Compared with [11] [12] [13], we discuss how the multi-received information affects a control system. Compared with [12] [13] [17], we investigate how the information importance affects a practical control system with the ML module. To the best of our knowledge, this is the first paper that designs the information importance for ML performance and verifies it in the real environment.

The main contribution of this paper is summarized as follows.

- By utilizing the information bottleneck principle, we propose to use the incremental feature as the metric to indicate the information importance. In particular, we prove that the proposed method is identical to the traditional method when the feature information is independent of the irrelevant data (See Lemma 1).
- We propose a method to calculate the incremental feature (See Lemma 2).
- We establish a prototype to verify the proposed metric by optimizing the packet management in a practical application scenario.
- Compared to the traditional method, the proposed packet management effectively reduces the communication traffic and in the meantime reduces the control error.

III. Framework and Proposed Metric

A. Teleoperation Framework

As shown in Fig. 2, we consider a typical teleoperation framework, where a local control system carries out a task according to the information that is updated from a remote source. Firstly, the packet is generated at the remote source, which is denoted as a variable $x \in X$. Then, the packet is updated to the local control system, and the latest delivered packet is denoted as $x' \in X$, which is relatively out of date compared to x in terms of the time delays and the update interval. Here, the update decision is charged by an update mechanism (policy) that quantifies the importance of the packet x compared to x' , determining whether x should be updated. Meanwhile, the control system utilizes x' and the control state $h \in H$ to generate control commands $y \in Y$ using a well-trained machine learning module, i.e., mapping process $p(Y|(X, H))$. This can be expressed by

$$y = (x', h) \odot p(Y|(X, H)). \quad (1)$$

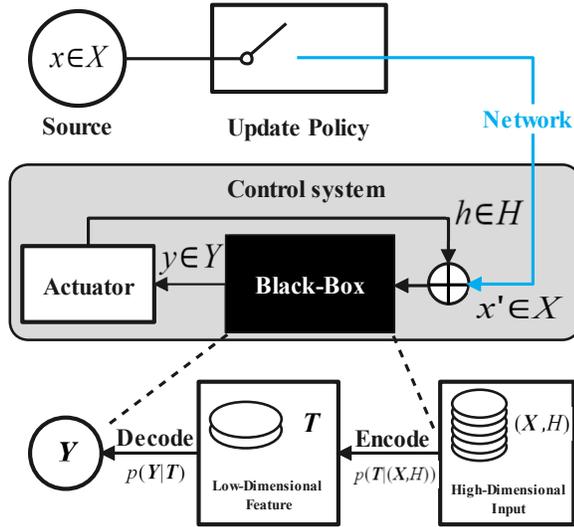


Fig. 2. A typical teleoperation framework with a machine learning module.

B. Information Bottleneck

Since $p(Y|(X, H))$ is achieved by the machine learning algorithm, it is a black box and difficult for analysis. Here, we open the black box using IB principle [18]. In particular, the machine learning module $p(Y|(X, H))$ can be explained as a cascade system that consists of both an encoder $p(T|(X, H))$ and a decoder $p(Y|T)$, i.e.,

$$p(Y|(X, H)) = p(T|(X, H)) \odot p(Y|T), \quad (2)$$

where

$$t = (x', h) \odot p(T|(X, H)), \quad (3)$$

$$y = t \odot p(Y|T). \quad (4)$$

In the about equations, t is the compressed feature information of (x', h) . Assuming $(X, H) \leftrightarrow T \leftrightarrow Y$ to be a Markov chain, the encoder $p(T|(X, H))$ and the decoder $p(Y|T)$ should be designed such that the feature t preserves most of the valuable information about y and discards the useless information in (x', h) . For the well-trained module $p(T|(X, H))$, the optimal encoder and decoder [16] [18] can be mathematically characterized by¹

$$\begin{aligned} & \{p^*(T|(X, H)), p^*(Y|T)\} \\ &= \underset{p(T|(X, H)), p(Y|T)}{\operatorname{arcm}} I((X, H); T) - \beta I(T; Y), \end{aligned} \quad (5)$$

where $I((X, H); T)$ indicates the compression rate, and $I(T; Y)$ is the relevant information, i.e., β is a trade-off parameter.

¹The mutual information between two random variables a and b is given by $I(a; b) = \int_{a \in A} \int_{b \in B} p_{a,b}(a, b) \log \frac{p_{a,b}(a, b)}{p_a(a)p_b(b)}$ [19].

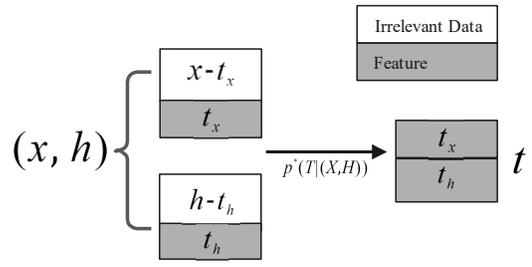


Fig. 3. feature extraction by the optimal encoder $p^*(T|(X, H))$.

C. Metric for Information Importance: Incremental feature

When we say that x is valuable compared to x' , it means the input information (x', h) is stale compared to (x, h) . In other words, (x, h) can provide incremental information compared to (x', h) . Additionally, because any input information (x', h) is going to be compressed to t' by Equ. (3), the actual difference between (x, h) and (x', h) for the control performance is indeed characterized by the difference between t and t' . To better illustrate this, we assume the valuable information hidden in the high-dimensional information (x', h) is denoted as $(t_{x'}, t_h)$, i.e., the irrelevant data is denoted as $(x' - t_{x'}, h - t_h)$. Then the encoder $p(T|(X, H))$ extracts the feature $(t_{x'}, t_h)$ from (x', h) . Considering an extreme case, if the incremental information from (x', h) to (x, h) does not contribute to any increment from $(t_{x'}, t_h)$ to (t_x, t_h) , updating x is valueless since it does not change the output y according to Equ. (4). Therefore, we propose a novel metric, called the incremental feature, to describe the value/importance of x .

The incremental feature indicates the difference between $(t_{x'}, t_h)$ and (t_x, t_h) . Compared to the traditional method that directly quantifies the difference between x' and x , the main relationship is summarized in Lemma 1.

Lemma 1: If we use the mutual information to quantify the information importance of the source information x compared to the latest delivered information x' , the information importance of low-dimensional feature $t_{x'}$ is identical to the one of high-dimensional information x' , when the feature information t_x is independent of the irrelevant data $x - t_x$. Mathematically, we have

$$I(t_{x'}; t_x) = I(x'; x) + H(n), \quad (6)$$

where $n \in N$ denotes the irrelevant data $x - t_x$, and $H(n)$ is the entropy of n , which is a constant.

Proof: See Appendix A.

However, it is difficult to evaluate the incremental feature from $t' = (t'_{x'}, t_h)$ to $t = (t_x, t_h)$, since h is unknown to the remote source. Even if there exists some special feedback link to transmit h , communication delays degrade performance. To address this, we propose to use $|t - t'|$ to evaluate the incremental feature, which then quantifies

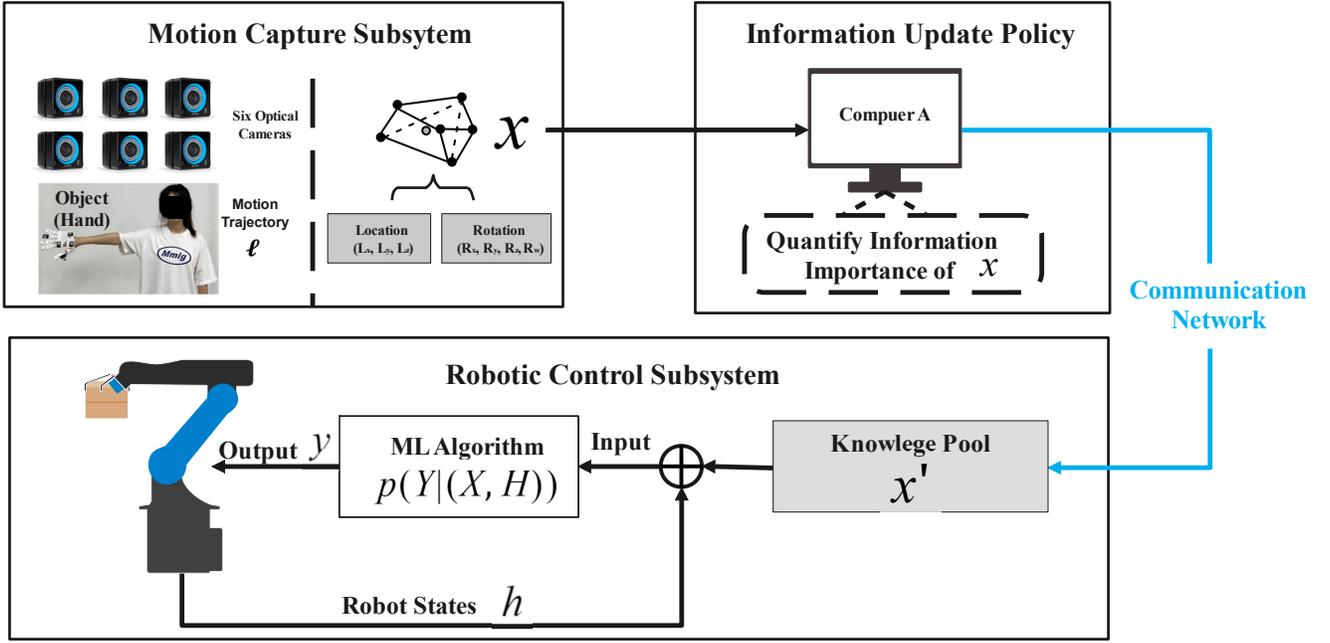


Fig. 4. System diagram of our motion-controlled teleoperation system.

the information importance of x , where $|t - t'|$ can be obtained by Lemma 2.

Lemma 2: If X and H are independent, we can use a constant control state h_0 to substitute the varying state h while calculating $t - t'$, i.e.,

$$\begin{aligned} t - t' &= (t_x, t_h) - (t'_x, t'_h) \\ &= (t_x, t_{h_0}) - (t'_x, t_{h_0}). \end{aligned} \quad (7)$$

Proof: See Appendix B.

Since h_0 is known to the source, the proposed metric can be expressed by

$$\begin{aligned} Dis(x', x) &= t - t' \\ &= |(x, h_0) \odot p^*(T|(X, H)) - (x', h_0) \odot p^*(T|(X, H))|. \end{aligned} \quad (8)$$

In the next section, we establish a prototype to the above metric in a practical application scenario.

IV. Experiment Design

In this section, we establish a prototype to verify the proposed metric $Dis(x', x)$. As follows, we first present the prototype with the update mechanism that quantifies the information importance based on the proposed metric. Then, we formulate an optimization problem based on the proposed metric. Finally, we use the Deep Q-network learning (DQN) method to solve the problem.

A. Established Prototype

1) Overall Architecture: As shown in Fig. 4, the established motion-controlled teleoperation prototype has three parts: the motion capture subsystem, the information update policy part, and the robotic control subsystem.

Firstly, the human's hand movement is the input of the entire system, which is captured by six optical cameras (OptiTrack prime 13). Then, Computer A with an information update policy determines when the sampled motion information should be updated to the robotic arm system. If the decision is to send a new update, the latest motion information will be transmitted to Computer B (controller) via a communication network. Based on the received information, Computer B generates control commands for the robotic arm (Franka Emika Panda). In this way, the robotic arm reproduces the human's hand movement in real time.

2) Motion Capture Subsystem: As shown in the upper left part of Fig. 4, the motion capture system consists of six optical-passive cameras driven by specific sensing data processing and management software called Motive. To capture the motion of the human's hand movement, the cameras are located at the ceiling corners of a 4 meter \times 4 meter area. The human's hand is attached with six reflective markers and then formulated as a rigid body. When the operator moves his/her hand with a trajectory ℓ , the capture system tracks the rigid body and obtains motion information m . m consists of both the coordinate location (L_x, L_y, L_z) and the quaternion rotation (q_x, q_y, q_z, q_w) of the rigid body. In particular, we use m_i to denote the sample of m at the time slot i . The system sample rate is $f_m = 120$ Hz, which indicates that m_i is generated with the same time interval $t_m = 1/f_m = 1/120$ s. Here, the packet x_i (Corresponding to X in Section III) at the time instant i consists of all

the generated samples, i.e.,

$$x_i = \{m_1, m_2, \dots, m_i\}. \quad (9)$$

3) Robotic Control Subsystem: As shown in the bottom part of Fig. 4, the robotic control subsystem consists of a controller (Computer B) and a robotic arm. Here, we use x'_i (Corresponding to X' in Section III) to represent all the delivered information from the 1st to the i -th time instant, i.e.,

$$x'_i = \{m_1 a_1, m_2 a_2, \dots, m_{j-1} a_{j-1}, m_j\}, \quad (10)$$

where m_j is the latest delivered information and $j \leq i$ due to the network delays. In the experiment, the robotic control is derived by the well-trained machine learning module $p(Y|(X, H)) = p(T|(X, H)) \odot p(Y|T)$ to achieve the real-time control. For notational simplicity, we use $\phi(\cdot)$ to denote the optimal encoder $p^*(T|(X, H))$. Firstly, the robotic arm uses $\phi(\cdot)$ to obtain the feature t_i , i.e.,

$$\begin{aligned} t_i &= (x_i, h) \odot p(T|(X, H)) \\ &\triangleq \phi(x_i, h). \end{aligned} \quad (11)$$

After that, the robotic arm uses the decoder $p^*(Y|T)$ to generate control commands y_i , i.e.,

$$y_i = t_i \odot p(Y|T), \quad (12)$$

which achieves real-time remote tracking control.

4) Information Update Policy: As shown in the upper right part of Fig. 4, Computer A runs the update policy. It determines the update decision a_i by quantifying the information importance of x_i compared to x'_i . Here, we use the following sequence to represent the update decisions for each motion information m_i ,

$$A = \{a_1, a_2, \dots, a_i\}. \quad (13)$$

If Computer A decides to send a new update in time slot i , denoted as $a_i = 1$, a communication packet carrying the latest information m_i is transmitted to the controller via a communication network. On the contrary, m_i will be discarded if $a_i = 0$, i.e., there is no information to be sent. Specifically, the update decision is made based on the proposed metric $Dis(x', x)$, i.e.,

$$a_i = D(Dis(x'_i, x_i)), \quad (14)$$

where $D(\cdot)$ is the update policy, i.e., a binary mapping function.

B. Problem Formulation

In the network, we adopt the same assumption that has been widely used [10] [11] [17]: The network can only service one update at the same time. In particular, if the decision $a_i = 1$, we use u_i to denote the time stamp when a new update is submitted. Then, each update occupies the communication network with a random time t_i , which

refers to real network delays. Thus, the server is idle only when

$$t_c \geq u_{i-1} + t_i, \quad (15)$$

where t_c indicates the current time instant. Otherwise, the server is busy.

Based on the previous analysis, an optimization problem is formulated as follows,

$$\min_{D(\cdot)} : D_{rms} = \lim_{N \rightarrow \infty} \sqrt{\frac{1}{N} \sum_{i=1}^N Dis(x'_i, x_i)^2}, \quad (16)$$

s.t.

$$Dis(x', x) = t' - t, \quad (17)$$

$$A = \{a_1, a_2, \dots, a_{N-1}\}, \quad (18)$$

$$x_i = \{m_1 a_1, m_2 a_2, \dots, m_{j-1} a_{j-1}, m_j\}, \quad (19)$$

$$a_i = \begin{cases} D(Dis(x'_i, x_i)), & t_c \geq u_{i-1} + t_i, \\ 0, & \text{otherwise.} \end{cases} \quad (20)$$

In the above problem, we use the proposed metric $D(Dis(x'_i, x_i))$ as the input of the decision function $D(\cdot)$. The goal is to minimize the information difference between the local control system and the remote source, i.e., keeping data fresh. It is challenging to solve the above problem since $\phi(\cdot) = p(T|(X, H))$ is the encoder in a machine learning module, which is a non-linear mapping. In addition, the distribution of service time cannot be accurately modeled since the real communication environment is time-variant. Therefore, we propose a machine learning solution in the next subsection.

C. Solution

In this section, we develop a deep q-learning network (DQN) method to find the optimal update policy $D(\cdot)$, where DQN interacts with our system. Here, we use three elements $\langle S, A, R \rangle$ to describe the proposed system. Firstly, the state sequences S consists of N states, i.e., $S = \{s_1, s_2, \dots, s_N\}$, where N represents the number of the state changes. Secondly, $A = \{a_1, a_2, \dots, a_N\}$ are the action sequences, which is identical to Equ. (13). $R = \{r_1, r_2, \dots, r_N\}$ is the reward sequences. Based on the state s_i and the action a_i , a reward result r_i returns.

In the time slot i , the state s_i is identical to the proposed metric $D(Dis(x'_i, x_i))$, which is the multi-dimensional incremental feature and obtained by Equ. (8). In the next time slot $i + 1$, the state $s_{(i+1)}$ is determined by s_i , a_i , since x_{i+1} and x'_{i+1} are determined by x_i , x'_i , and a_i . Additionally, the reward function for the action a_i under a specific state s_i is expressed as

$$r(s_i, a_i) = \begin{cases} -1, & \text{if } a_i = 1 \text{ and } t_c < u_{i-1} + t_i, \\ -Dis(x_i, x'_i)^2, & \text{otherwise.} \end{cases} \quad (21)$$

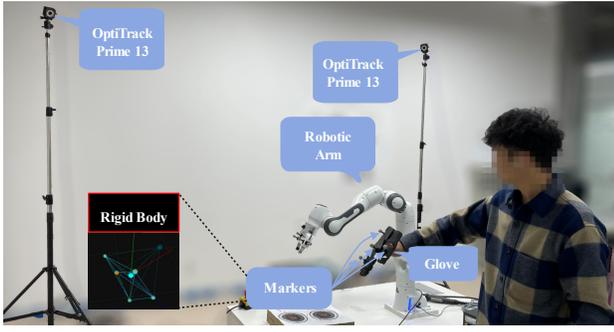


Fig. 5. Prototype of our motion-controlled teleoperation system.

In the above equations, we use -1 to denote an irrational decision: It returns an update decision while the server is busy, i.e., $a_i = 1$ for $t_c < u_{i-1} + t_i$. In other cases, $r(s_i, a_i)$ is equal to $-Dis(x_i, x'_i)^2$. The DQN method maximizes

$$Q(s_i, a_i) = \underbrace{r(s_i, a_i)}_{\text{Current reward}} + \eta \underbrace{\text{Max}\{r(s_{i+1}, a_{i+1} = 0), r(s_{i+1}, a_{i+1} = 1)\}}_{\text{Future reward}}, \quad (22)$$

where the former item indicates the instant benefit from the current decision and the latter item indicates the maximum expected benefit can be obtained under future states, i.e., the future benefit. In other words, It actually maximizes the long-term average reward, which is identical to the object function of the original problem. Then, an equivalent optimization problem becomes

$$\min_{Q(\cdot)} : H = \lim_{N \rightarrow \infty} \sqrt{\frac{1}{N} \sum_{i=1}^N r(s_i, a_i)}, \quad (23)$$

s.t.

$$s_i = \{x'_i, m_i\}, \quad (24)$$

$$a_i = \begin{cases} 0, & \text{if } Q(s_i, a_i = 1) < Q(s_i, a_i = 0), \\ 1, & \text{otherwise.} \end{cases} \quad (25)$$

In the above problem, H is equal to the original objective function D_{rms} . Then, $Q(s_i, a_i = 1) < Q(s_i, a_i = 0)$ indicates the action $a_i = 0$ is better than the action $a_i = 1$ for a long-term benefit when the state is s_i , i.e., it is identical to $D(Dis(x, x') = 0)$ of the original problem. Otherwise, it indicates $a_i = 1$, which is identical to $D(Dis(x, x') = 1)$. As a result, the original problem is solved by optimizing $Q(\cdot)$.

V. Experiment Setting and Results

A. Experiment Setting

To evaluate the proposed approach for optimizing the real-time teleoperation system, we conduct practical experiments to provide real data for training and testing.

As shown in Fig. 5, an operator can see the robot in person, which allows he/she to do interactive control. The operator constantly moves one hand and draws a “W” shape trajectories in the air. The coordinate and gestation data are captured in the Optitrack motion tracking system. In every 8 ms, the motion tracking system will update its collected data. In total, 1800-seconds with 216,000 samples was obtained in the experiment. In particular, the samples are divided into two parts: Training data-set with 200,000 samples and test data-set with 16,000 samples. Then, the training data is given to the DQN. The overall iteration number of DQN is 100, where the training data samples are used to optimize Q-network in each iteration. Additionally, the memory size of DQN is set as 1000, which caches the last 1000 state transform processes. Then, 128 random batches extracted from the memory are used to accelerate the optimization by breaking the correlation between consecutive samples. As a result, there are $N = 100 \times 200000 \times (128 + 1) \approx 1.3 \times 10^9$ samples used for training.

The network is set with different maximum network delays. In particular, we assume that the maximum time delay is 0 ms, when all devices are directly connected via wired cables, i.e., no network communication. In other cases, we use a virtual private network (VPN) to serve the information update and set the maximum time delay as [50, 100, ..., 400] ms by tuning VPN parameters. Additionally, the network cost is investigated by the server occupation rate, which is defined as

$$S = \mathbb{E}\left\{\frac{T_{busy}}{T_{idle} + T_{busy}}\right\}, \quad (26)$$

where T_{busy} is the overall duration when the server is busy, and T_{idle} is the overall duration when the server is idle. Then, a policy with a lower S can reduce network traffic.

B. Experiment Results

For comparison, the zero-wait policy is treated as the baseline.

- Zero-wait policy (baseline): In this update policy, a new packet is submitted once the previous packet is delivered to the control system.
- Proposed policy: Based on the proposed update policy $D(\cdot)$, the update decision a_i is obtained by Equ. (20).

It is worth to note that traditional methods such as ϵ -wait in [11] are not adopted as the baseline in this paper, since these methods are developed in an ideal model, which can not be directly applied in a real environment.

In this subsection, control performance is evaluated by the root mean square importance of both the proposed metric $Dis(x, x')$ and the tracking error. $Dis(x, x')$ is defined in Equ. (8). The real tracking error is the distance between the robotic arm and the human hand in real time, where an example is shown in Fig. 6. During the time $[t_0, t_0 + 5s]$, the red curve indicates the trajectory of the human hand while the blue curve indicates the trajectory

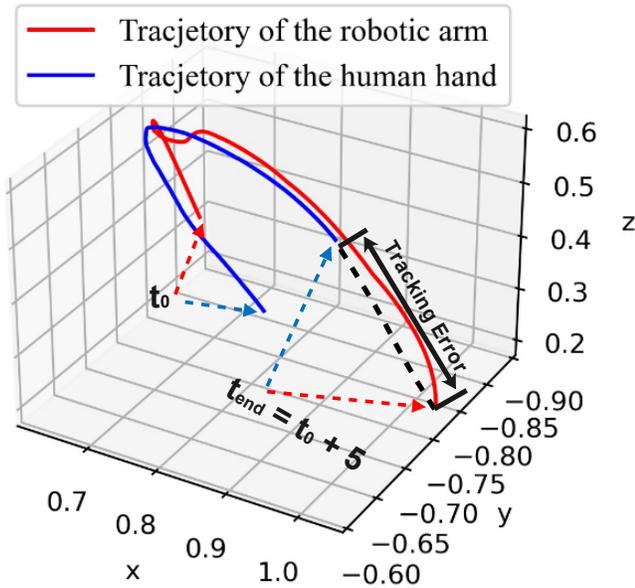


Fig. 6. Tracking error obtained by the motion capture system (Optitrack cameras) in real time.

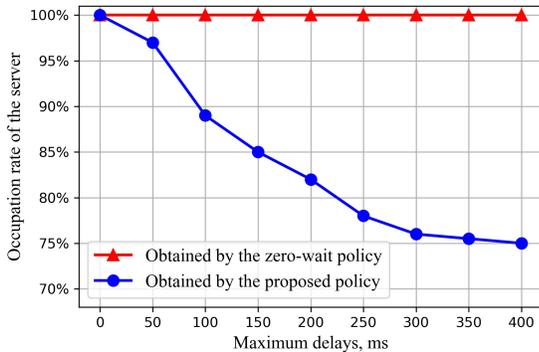


Fig. 7. Network communication cost under different communication conditions.

of the robotic arm, which are both captured by the motion capture system.

When the maximum delay is 0 ms, as shown in Fig. 7, the server occupation rates under both policies is 100%, i.e., the proposed method is equivalent to the zero-wait policy in this case. This is because redundant information does not lead to congested network traffic when the maximum delay is 0ms. Additionally, as shown in Fig. 8, the incremental feature D_{rms} (Blue and red curves) is 0, since the controller has all information about the source. However, in this case, the tracking error (Black and green curves) still exists because of physical limits of the robotic arm, e.g., there exist upper bonds of velocity and acceleration for each joint of the robotic arm.

When the maximum delay increases, as shown in Fig. 7, the proposed method effectively reduces network traffic up to 25%. Meanwhile, it also reduces the estimation error

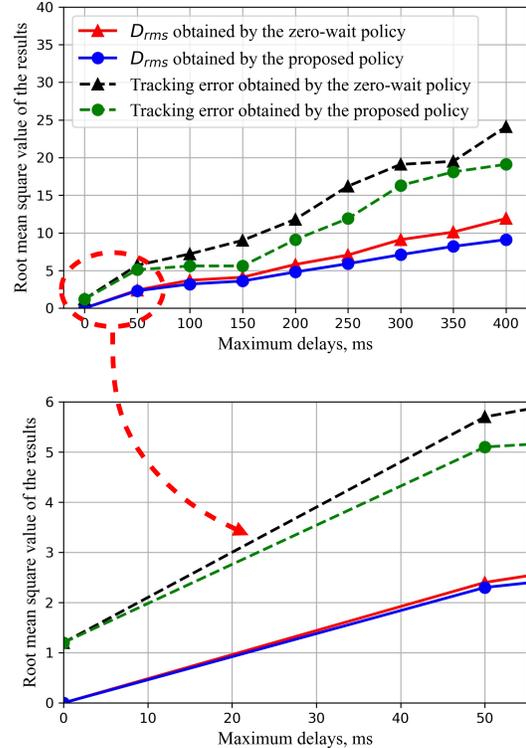


Fig. 8. Estimation error and tracking error under different communication conditions.

Δm_i as shown in Fig. 8, e.g., E_{rmse} is reduced from 11.9 cm to 9.1 cm (23.5%), and the tracking error is reduced from 24.1 cm to 19.1 cm (20.7%) when the maximum delay is 400 ms.

VI. Conclusion

In this paper, we firstly established a feature-oriented teleoperation framework to quantify the packet importance, which was derived from the information bottleneck principle. After that, a packet management method was proposed to increase the average feature quantity of the receiver. Additionally, we built a prototype for verifying the proposed method, where sensing, communication, and robotic control were included. In the experiment, the proposed information update policy effectively reduced the communication traffic up to 25% and in the meantime reduced the robotic tracking error up to 23.5%.

Appendix A

Proof of Lemma 1

If $x - t_x$ is independent to x , we use n to denote $x - t_x$ for simplicity. Then, we have

$$p_{t_x, n}(t_x, n) = p_{t_x}(t_x)p_n(n). \quad (27)$$

As shown in Equ. (28), we complete the proof.

$$\begin{aligned}
I(x; x') &= I((t_x, n); (t_{x'}, n)) \\
&= \int_{(t_x, n) \in (T_x, N)} \int_{(t_{x'}, n) \in (t_{x'}, N)} p_{t_x, t_{x'}, n}((t_x, n), (t_{x'}, n)) \log \frac{p_{t_x, t_{x'}, n}((t_x, n), (t_{x'}, n))}{p_{t_x, n}(t_x, n) p_{t_{x'}, n}(t_{x'}, n)} \\
&= \int_{t_x \in T_x} \int_{t_{x'} \in t_{x'}} \int_{n \in N} p_{t_x, t_{x'}, n}(t_x, t_{x'}, n) \log \frac{p_{t_x, t_{x'}, n}(t_x, t_{x'}, n)}{p_{t_x}(t_x) p_{t_{x'}}(t_{x'}) p_n(n)^2} \\
&= \int_{t_x \in T_x} \int_{t_{x'} \in t_{x'}} \int_{n \in N} p_{t_x, t_{x'}, n}(t_x, t_{x'}, n) \log \frac{p_{t_x, t_{x'}}(t_x, t_{x'})}{p_{t_x}(t_x) p_{t_{x'}}(t_{x'}) p_n(n)} \\
&= \int_{t_x \in T_x} \int_{t_{x'} \in t_{x'}} p_{t_x, t_{x'}}(t_x, t_{x'}) \log \frac{p_{t_x, t_{x'}}(t_x, t_{x'})}{p_{t_x}(t_x) p_{t_{x'}}(t_{x'})} \int_{n \in N} p_n(n) - \int_{t_x \in T_x} \int_{t_{x'} \in t_{x'}} p_{t_x, t_{x'}}(t_x, t_{x'}) \int_{n \in N} p_n(n) \log p_n(n) \\
&= I(t_x, t_{x'}) + H(n).
\end{aligned} \tag{28}$$

$$\begin{aligned}
t - t' &= (x \odot p(T_x|X), h \odot p(T_h|H)) - (x' \odot p(T_x|X), h \odot p(T_h|H)) = (x \odot p(T_x|X) - x' \odot p(T_x|X), 0) \\
&= (x \odot p(T_x|X) - x' \odot p(T_x|X), h_0 \odot p(T_h|H) - h_0 \odot p(T_h|H)) = (x, h_0) \odot p^*(T|(X, H)) - (x', h_0) \odot p^*(T|(X, H))
\end{aligned} \tag{31}$$

Appendix B Proof of Lemma 2

Since $x \in X$ and $h \in Y$ are independent, their feature information t_x and t_h are also independent, i.e., t can be expressed by $t = (t_x, t_h)$. Define a mapping function $p(T_x|X)$ and $p(T_x|X)$ which satisfy

$$t_x = x \odot p(T_x|X), t_h = h \odot p(T_h|H), \tag{29}$$

and we have

$$\begin{aligned}
t &= (x, h) \odot p^*(T|(X, H)) \\
&= (t_x, t_h) \\
&= (x \odot p(T_x|X), h \odot p(T_h|H))
\end{aligned} \tag{30}$$

Then, we obtain Equ. (31).

References

- [1] M. Cintuglu, O. Mohammed, K. Akkaya, and A. Uluagac, "A survey on smart grid cyber-physical system testbeds," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 446-464, 2017.
- [2] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez et al., "Real-time wireless sensor-actuator networks for industrial cyber-physical systems," in *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1013-1024, May 2016.
- [3] G. Yang, L. Xie, M. Mäntysalo, X. Zhou, Z. Pang, et al., "A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box," in *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2180-2191, Nov. 2014.
- [4] G. Yang, Z. Pang et al., "Guest editorial: Enabling technologies in health engineering and informatics for the new revolution of healthcare 4.0," in *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 9, pp. 2442-2443, Sept. 2020.
- [5] S. Chen, J. Hu, Y. Shi et al., "Vehicle-to-everything (V2X) services supported by LTE-based systems and 5G," *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70-76, 2017.
- [6] H. Lv, G. Yang, H. Zhou, X. Huang, H. Yang, and Z. Pang, "Teleoperation of collaborative robot for remote dementia care in home environments," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 8, pp. 1-10, no. 1400510, 2020.
- [7] S. Li, D. Xu, and S. Zhao, "5G Internet of Things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1-9, 2018.
- [8] Z. Pang, L. Zheng, J. Tian, S. Kao-Walter, E. Dubrova, Q. Chen, "Design of a terminal solution for integration of in-home health care devices and services towards the Internet-of-Things", *Enterprise Information Systems*, vol. 9, no. 1, pp. 86-116.
- [9] Y. Sun and B. Cyr, "Sampling for data freshness optimization: Non-linear age functions", *Journal of Communications and Networks*, vol. 21, no. 3, pp. 204-219, 2019.
- [10] Y. Sun, E. Uysal-Biyikoglu, R. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," in *Proc. IEEE INFOCOM 2016*, San Francisco, CA, 2016, pp. 1-9.
- [11] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or Wait: How to keep your data fresh," *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492-7508, Nov. 2017.
- [12] T. Soleymani, J. S. Baras, and K. H. Johansson, "Stochastic control with stale information—part I: Fully observable systems," in *Proc. IEEE 58th Conference on Decision and Control (CDC)*, Nice, France, 2019, pp. 4178-4182.
- [13] Ayan O, Vilgelm M, Klügel, Markus et al., "Age-of-information vs value-of-information scheduling for cellular networked control systems", *International Conference on Cyber-Physical Systems*, Montreal, QC, April 16–18, 2019.
- [14] X. Tong, L. Li, G. Zhao, B. Chang, and Z. Chen, "Beyond Fresh Update: Packet Management for Real-Time Feedback Control," in *Proc. IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, London, UK, pp. 1-6, 2020.
- [15] X. Tong, L. Li, G. Zhao, Z. Meng, Z. Chen, and G. Yang, "Packet management for optimizing control performance in real-time feedback control systems," *46th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, 2020, pp. 4362-4367.
- [16] N. Tishby, F. C. Pereira, and W. Bialek "The information bottleneck method", *arXiv preprint physics/0004057*, 2000.
- [17] Y. Sun and B. Cyr, "Information aging through queues: A

- mutual information perspective,” in proc. IEEE SPAWC 2018, Kalamata, 2018, pp. 1-5.
- [18] R. Shwartz-Ziv and N. Tishby, “Opening the black box of deep neural networks via information”, arXiv preprint/1703.00810, 2017.
- [19] T. M. Cover and J. A. Thomas, Elements of Information Theory, 2nd ed. John Wiley & Sons, 2006.