An End To End Tactile Cyber Physical System Design

Arjun N, Ashwin S M, Kurian Polachan, T V Prabhakar, and Chandramani Singh
Department of Electronic Systems Engineering
Indian Institute of Science, Bangalore
Email: arjun,ashwin,kurian,tvprabs,chandra@iisc.ac.in

***Abstract*—We describe design and implementation of an end to end *Tactile Cyber Physical System* (TCPS). We begin with proposing a generic architecture for TCPSs. We then describe a typical system in which a robotic arm at one end of the network replicates the posture of a human hand at the other end with a tactile glove. For the human operator end, we design a tactile glove having Inertial Measurement Unit (IMU) sensors for motion capture and vibrotactile actuators for tactile feedback. On the teleoperator end, we use a PhantomX reactor robotic arm with force sensors and design an inverse kinematic engine for this. We also provide a mechanism for visual feedback using a USB camera on the teleoperator end. We describe in detail the hardware and software designs of the tactile glove and the controller board for the robot. We also detail the industrial design of the tactile glove. We perform detail experiments to characterize the tactile glove, robotic arm and round-trip delay of the TCPS implementation.**

***Index Terms*—tactile internet; tactile glove; tactile cyber physical system;**

I. Introduction

Internet Traffic such as web, packet voice, online chats, games and file transfer are all human triggered. In recent years, with the advent of the Internet of Things (IoT), data is also being generated and consumed by machines without human intervention. In fact, machine type communication is stipulated to dominate the Internet traffic in future. The IoT brings collective intelligence that aims at improving quality of our lives. For instance, ambient intelligence builds on several underlying concepts such as pervasive computing, ubiquitous computing, context awareness etc. Contributing to this IoT space is yet another fast-emerging class of Internet traffic, known as human-centric or human in the loop communication. This traffic is generated by humans and human controlled application-specific robots to suit each other's consumption, so this should meet the requirements of real-time control interactions. In particular, this traffic is characterized by ultra-low latency (round trip times of the order of 1 ms) and ultra-high-reliability (failure rates of the order of 10^{-9}). Communication with such ultra-high *Quality of Experience* (QoE) could now be a possibility thanks to emerging mobile Internet or cellular Internet technologies. For example, the upcoming 5th generation (5G) cellular systems are expected to offer a round-trip latency of the order of a few ms as compared to 30-50 ms offered by today's fastest Long Term Evolution (LTE) systems. While LTE based networks are limited to application domains such as cloud computing, real-time games and packet voice, 5G networks will also suit ultra-low-latency applications involving virtual and augmented reality. Here, humans will be in a position to interact with remote machines and objects as if they are manipulating them locally. The requirement of ultra-low latency and failure rates differentiate these so-called Tactile Cyber Physical Systems (TCPS) from the well known networked control systems where advance delay compensation techniques suffice to circumvent lossy and delay-prone links.

With tactile feedback available to provide cutaneous, kinesthetic and haptic feedback, TCPS applications can include telepresence, telemedicine, connected cars, smart grids and a variety of other interactive applications involving sensors and actuators. All of these applications involve an operator remotely controlling and steering an object in one direction and receiving tactile and visual feedback to support seamless operation in the opposite direction. TCPSs put forward many new challenges and wish specifications for designing a communication infrastructure, called Tactile Internet, that can ensure a few ms round trip delay and high-grade carrier robustness [1], [2], [3]. However, the challenges are not limited to the network subsystem. They also include the architectural and design aspects of a holistic TCPS. These include TCPS infrastructure components and sub-components, e.g., high-speed kinematic devices to capture operators' actions and haptic devices to apply haptic feedback.

Our motivation is to study the challenges underlying TCPS design and issues such as maintaining latency within the stipulated budget in each of the system components. Towards this, we design and develop an end-to-end system that is minimalist yet incorporates most of the crucial features of a typical TCPS. We assume the availability of Tactile Internet and focus on the architectural and design aspects of realizing a TCPS. We also invest effort in designing and fabricating a cost-effective and functionally complete tactile glove, that can capture the kinematic actions of the operator's hand and can also use the haptic information passed to it to emulate haptic feedback. Transmission and processing of control parameters and feedback happen in real time. Observe that there is an interplay between control and Internet in our work. The control parameters associated with the kinematic and inverse kinematic equations at the operator and teleoperator ends are transported over the network link.

The remainder of the paper is organized as follows. We list the related work in Section I-A. In Section II, we present a generic architecture for TCPS end systems. Here we also detail the building blocks of the operator and the teleoperator ends of the proposed TCPS. We describe the design of the two end systems and of the connecting network in Section III.

978-1-5386-7468-0/18/$31.00 ©2018 IEEE
DOI 10.1109/EITEC.2018.00007

We report the measurement results in Section IV. Finally, in Section V, we conclude the paper.

A. Related Work

Tactile Internet and TCPS being fairly new concepts, most of the work in this area is still limited to research in universities [1][2][3], and much of this work revolves around the design aspects of Tactile Internet. Majority of the Tactile Internet literature focuses on latency and reliability of Tactile Internet links. There has been some work on design and implementation aspects of specific TCPS components. For instance, researchers have used Inertial Measurement Unit (IMU) sensors to capture human arm and body movements [4], [5], [6], [7]. Also, various approaches have been proposed for tactile feedback in human interactions [8], [9]. Specific to tactile gloves, there exists few commercial products [10], [11]. These gloves have mechanisms for motion capture and tactile feedback. In Table 1, we present costs and features of a few of these tactile gloves and compare these to the cost and the feature set of our design.

Recently, there has been some work on tactile communication systems with haptic feedback [12], [13]. However, in these works, the two ends are connected through a USB or through RF, not through a network as in our setup.

II. Architecture and Building Blocks

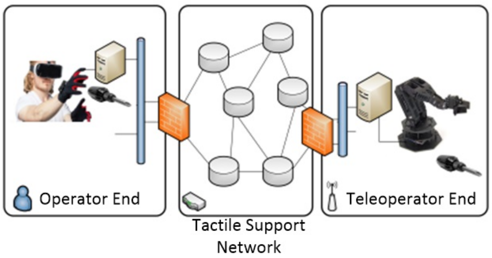

Fig. 1: High level block diagram of TCPS

We present an overview of the envisaged TCPS in Figure 1. The system consists of three subsystems: (a) an operator end, (b) a teleoperator end and (c) tactile support network. Following are various elements in each of these three subsystems:

1) *At the Operator End:*
   a) A human operator wearing a tactile glove equipped with motion capture elements and tactile feedback actuators. The motion capture elements in the tactile glove help to capture human arm and finger motion. This information is used to control the teleoperator to do the desired task.
   b) A visual feedback mechanism to provide visual feedback about the teleoperator end.
2) *At the Teleoperator End:* A robotic arm equipped with force sensors and a camera. Force sensors and camera provide tactile and visual feedback, respectively, to the human operator.
3) *Tactile Support Network:* An infrastructure to transmit actuation signals from the operator end to the teleoperator end and tactile and visual feedback signals in the reverse direction. This network is intended to support very low round-trip latency and to provide very high reliability.

Our primary focus in this work is on the design of the operator and the teleoperator ends. We assume that a network infrastructure is already in place to support communication between the two ends.

Now, we present detail description of the elements of the two end-subsystems.

A. Operator End

1) Tactile Glove: The tactile glove provides a natural interface for the human operator to control robotic arm based teleoperators. The tactile glove captures motion of the human arm and fingers, and at the same time, provides tactile feedback about teleoperator's actions to the human operator. In the tactile glove, we require kinematic sensors for motion tracking, haptic actuators for applying feedback, and a controller for reading/processing sensor data, driving actuators and communicating with a tactile support network.

In this section, we briefly cover different components of the glove and their design. Please see [14] for an elaborate discussion.

a) *Motion Tracking:*
   i) *IMU Sensors and Body Frame Axes:* For tracking hand motion, we use five MEMS-based IMUs that have 9-axis degrees of freedom. These are sourced from TDK-Invense [15] (see Figure 2). The orientation axes or the sensor body frame axes is defined during sensor fabrication and available from the datasheets.
   We use IMU sensors because they strike a better balance between accuracy, sampling rate, cost and compactness than other available kinematic sensors. For instance, optical sensors are accurate, but they suffer from occlusion, impacting ease of use. Moreover, they are expensive and also need an additional camera, increasing the product component count. Since we use IMUs to track limited hand postures only, the lack of accuracy does not impact much. We also use the complementary filtered DCM algorithm to compensate for the lack of accuracy of IMUs [16].

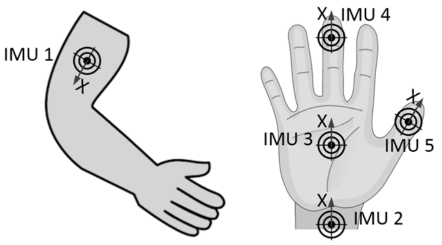

Fig. 2: Placement and orientation of the five IMU sensors located on the hand and the palm. The arrow labelled *X* denotes the sensor body frame x-axis direction of the corresponding IMU sensor.

   ii) *Global Frame Axes and DCM:* This axes forms the reference for the body frame axes. The global frame orthogonal axes I, J, K are determined by earth's magnetic and gravitational field vectors [17] and computed in the form of a Direction Cosine Matrix (DCM) [18]. (1) represents the relation between unit vectors through 3 × 3 DCM, where i,j and k are unit vectors of sensor body frame axes.

$$\begin{bmatrix}\mathbf{I}\\\mathbf{J}\\\mathbf{K}\end{bmatrix}=\begin{bmatrix}cos(\mathbf{I},\mathbf{i}) & cos(\mathbf{I},\mathbf{j}) & cos(\mathbf{I},\mathbf{k})\\cos(\mathbf{J},\mathbf{i}) & cos(\mathbf{J},\mathbf{j}) & cos(\mathbf{J},\mathbf{k})\\cos(\mathbf{K},\mathbf{i}) & cos(\mathbf{K},\mathbf{j}) & cos(\mathbf{K},\mathbf{k})\end{bmatrix}\begin{bmatrix}\mathbf{i}\\\mathbf{j}\\\mathbf{k}\end{bmatrix} \tag{1}$$

Magnetometer Calibration: To overcome the non-ideal response surfaces, the offsets along i, j, and k are subtracted from subsequent magnetometer readings. This step is known as hard iron bias correction and makes the response surface of magnetometer spherical [19]. This test should ensure that there is no external magnetic field in the vicinity.

iii) *Computing DCM:* While the first and third row of the DCM of an IMU sensor may be populated from its accelerometer and magnetometer readings, the second row can be computed as the cross product of the third and first rows (see (2)). Owing to their noisy readings, angular velocities from gyroscope readings are also used for estimating the DCM. To compensate for drift errors in DCM computation, complimentary filtered DCM algorithm is applied [16].

$$\mathrm{DCM}=\begin{bmatrix}m_x & m_y & m_z\\l_x & l_y & l_z\\a_x & a_y & a_z\end{bmatrix},\ \begin{bmatrix}l_x\\l_y\\l_z\end{bmatrix}=\begin{bmatrix}a_x\\a_y\\a_z\end{bmatrix}\times\begin{bmatrix}m_x\\m_y\\m_z\end{bmatrix} \tag{2}$$

iv) *The Tracking Algorithm:* To track the wrist position, as shown in Figure 2, IMU1 and IMU2 are mounted such that their body frame axes unit vectors i are aligned in the direction of the arm length. If the arm length L_u and forearm length L_f are known, we can determine the wrist position in the global frame using DCM1 and DCM2 (see (3)).

$$\begin{bmatrix}X_g\\Y_g\\Z_g\end{bmatrix}=\mathrm{DCM1}\begin{bmatrix}L_u\\0\\0\end{bmatrix}+\mathrm{DCM2}\begin{bmatrix}L_f\\0\\0\end{bmatrix} \tag{3}$$

To track the palm orientation, IMU2 and IMU3 are used and requires their unit vector i to be aligned to the hand finger. The pitch θ and yaw ϕ is then computed using DCM2 and DCM3 (see (4)).

$$\mathrm{DCMp}=\mathrm{DCM2}^{\mathrm{T}}\mathrm{DCM3} \tag{4a}$$

$$\theta=-\arcsin(\mathrm{DCMp}(3,1)) \tag{4b}$$

$$\phi=-\arctan\left(\frac{\mathrm{DCMp}(2,1)}{\mathrm{DCMp}(1,1)}\right) \tag{4c}$$

To track finger movement, IMU4 and IMU5 are used. They track middle finger and thumb respectively. Their body frame unit vectors k are parallel when the palm is open and perpendicular when the two fingers touch each other. We use DCMf(3,3), the cosine of the angle between body frame unit vectors k of IMU4 and IMU5 to track finger movement. Here DCMf is computed using (5).

$$\mathrm{DCMf}=\mathrm{DCM4}^{\mathrm{T}}\mathrm{DCM5}. \tag{5}$$

v) *Transformation to User Frame Axes:* Since wrist positions are usually specified in user frame axes, a calibration step from global frame to user frame axes is required. For this, we use (6). DCMc in the equation is the DCM1 computed during the power-up sequence. DCMc captures the relative orientation of user frame axes with reference to global frame axes.

$$\begin{bmatrix}X_u\\Y_u\\Z_u\end{bmatrix}=\mathrm{DCMc}^{\mathrm{T}}\begin{bmatrix}X_g\\Y_g\\Z_g\end{bmatrix} \tag{6}$$

b) *Tactile Feedback Actuators:* We provide tactile feedback to the glove in the form of vibration. We use a *eccentric rotating mass* (ERM) motor to produce this vibration in the glove. An ERM motor is a DC motor having an off-center mass on its shaft. It vibrates when a DC voltage is applied across it. Such an actuator is called a *vibrotactile actuator*. We can control the vibration amplitude through the voltage applied to the motor. Varying the motor drive voltage in accordance with the force sensed by the robotic arm allows us to give tactile feedback to the human operator in the form of vibration whose amplitude is modulated by the sensed force. We use "Vibrating Mini Motor Disc" from Adafruit [19] as the ERM motor in our glove. We place two vibrotactile actuators, one on the thumb and another on the index finger. These are controlled by the force sensed by the robotic arm.

Though ERM motors are less precise compared to Linear Resonant Actuators (LRA) and Piezoelectric Actuators (PA), driving them requires low voltage DC signals only. On the other hand, LRAs need AC signal and power amplifiers and PAs need high voltage DC signals. These would increase the component count and the driver circuit cost.

c) *Controller:* This controller does all the processing in the glove. We have chosen Programmable System on Chip (PSoC) 5LP, a system on chip from Cypress Semiconductor, as the central controller [20]. This controller is configurable and allows quick design changes in both analog and digital domains.

d) *Communication Interface:* We have designed the glove to have a wired interface. We use Universal Asynchronous Receiver Transmitter (UART) for the wired interface. The glove can use this interface to communicate to a computer (PC). We use wired LAN to send the glove data from the PC to the teleoperator side to control the robotic arm. We also use this channel to communicate tactile feedback to the glove from the teleoperator side.

e) *Power Supply:* The glove needs a power source to power-up PSoC, IMU sensors and tactile feedback actuators. Rechargeable batteries are a good option to power the glove. Lithium-Polymer (Li-Po) batteries are a popular class of rechargeable batteries [21]. We use a 1200 mAh Li-Po cell to power the glove. Apart from the battery, two other important modules in the power supply section are voltage regulator and battery charger IC. We regulate the battery voltage to 3.3V to power PSoC, IMU sensors and vibrotactile actuators. Our glove is designed to be charged through USB.

2) Visual Feedback: In the proposed system, we use a PC in front of the operator to provide visual feedback.

B. Teleoperator End

Teleoperator end of our TCPS consists of a robotic arm whose movement is controlled by the human operator over a network. The robotic arm is equipped with force sensors for tactile feedback and a 2-D camera for visual feedback. We use an off the shelf PhantomX as the robotic arm [22], [23]. It has 5 degrees of freedom with a custom parallel gripper as an end effector.

The teleoperator end consists of the following elements.

1) *Inverse Kinematics Engine:* Before describing our inverse kinematics engine we introduce the notion of robot kinematics. Robot kinematics refers to the analytical study of the motion of a robot manipulator. Robot kinematics can be divided into forward kinematics and inverse kinematics

   a) Forward kinematics refers to the use of kinematics equations to compute the position of the end effector of a robot manipulator from the given parameters (link lengths and joint angles) of the robot.

   b) Inverse kinematics refers to use of kinematics equations of a robot to determine the parameters that yield the desired position of the end effector. More precisely, it determines the joint angles since the link lengths are fixed for a robot.

   We need an inverse kinematics engine at the teleoperator end to help the robot manipulator track arm motion of the human operator at the other end. The inputs to our inverse kinematic engine are the position of the human operator's wrist and orientation of palm, both obtained from the tactile glove. The engine outputs the joint angles of all the joints The exact number of outputs depends on degrees of freedom of the robot manipulator.

   Our PhantomX robot has 5 degrees of freedom (i.e., 5 joint angles). The first three joint angles determine the position of the end-effector. The remaining two, pitch joint angle and gripper joint angle, determine the orientation of the end-effector. For a PhantomX robot, the input to the inverse kinematics equations is operator‘s wrist position and palm orientation. Inverse kinematic uses the wrist position to determine the first three joint angles of the end-effector. Pitch and gripper joint angles are directly read off from the supplied pitch of palm with respect to forearm and orientation and finger tracking information (tips of the thumb and middle finger).

   Let (X, Y, Z) denote the input wrist position. Let θ_1, θ_2 and θ_3 denote the desired first three joint angles and l_1, l_2 and l_3 denote the link lengths of the robot. Then inverse kinematics entails determining $\theta_1, \theta_2, \theta_3$ corresponding to X, Y, Z. We illustrate the relationship among these variables in the kinematic diagram of our robot (see Figure 3).

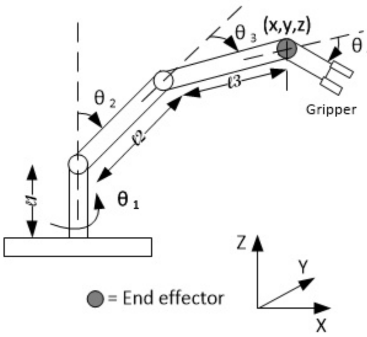

Fig. 3: Kinematic diagram of PhantomX reactor robotic arm

   From the front view of the kinematic diagram, we can write the following equations connecting X, Z and joint angles and link lengths.

$$Z = l_1 + l_2 \cos\theta_2 + l_3 \cos(\theta_2 + \theta_3) \tag{7}$$

$$X = (l_2 \sin\theta_2 + l_3 \sin(\theta_2 + \theta_3)) \cos\theta_1 \tag{8}$$

   Similarly, from the top view of the kinematic diagram, we can write the following equation connecting X, Y and joint angles and link lengths.

$$Y = (l_2 \sin\theta_2 + l_3 \sin(\theta_2 + \theta_3)) \sin\theta_1 \tag{9}$$

   From (7)-(9), we derive an analytical expression for θ_1, θ_2 and θ_3 in terms of X, Y, Z. We use this analytical expression to set the first three joint angles. 4^{th} joint angle θ_4 is set with the palm pitch information. We use DCMf(3,3) that tracks the finger movement to set the rotary gripper joint.

2) *Force Sensing System:* We attach two force sensors to the two wings of the end manipulator's gripper to provide force feedback about its tactile interactions. We use piezoresistive force sensors from Parallax for this purpose [24]. A piezoresistive sensor acts as a variable resistor in an electrical circuit. When we do not exert any force on the sensor, its resistance is very high (greater than 5MΩ). On the other hand, when a force is applied to the sensor, the resistance decreases to the range of 10-20KΩ

   Both capacitive and resistive methods are widely used for sensing force. Resistive methods have the advantage that their sensing circuits are fairly simple and their response is faster. In resistive methods, piezoresistive sensors are preferred owing to their compactness and low cost.

3) *Controller:* This controller does all the processing at the teleoperator end. For the sake of symmetry in design with the operator end, we choose the same PSoC 5LP as the central controller as used in the tactile glove at operator side.

III. System Design

A. Tactile Glove Design

This includes hardware and software design of the tactile glove.

1) Hardware Design: Hardware design of the glove involves the designs of main processor board of the glove and IMU boards of the sensors.

a) *Main Processor Board:* The main processor board houses the central controller, PSoC 5LP. We use the PWM module inside PSoC to control the ERM vibrotactile actuators. The main processor board also houses the voltage regulator, the battery charger IC, the driver circuit needed to drive the ERM motors and a few control switches and indication LEDs.

b) *IMU Board:* The IMU board houses the MPU-9250 IMU sensor which communicates with PSoC over I2C protocol. We need five IMU boards, one for each IMU.

2) Software Design: We first power up the glove to initialize the PSoC and MPU-9250 modules. Upon power-up, a calibration routine is run to compute DCMc. Once done with calibration, the software enters an infinite while loop. Every 10 ms, a timer inside PSoC generates an interrupt. At each such epoch, the IMU values are read and wrist position, palm orientation and fingers' locations are computed. These measurements are sent to a PC through a UART link. Also, the force sensor measurements we receive from the PC are used to control the ERM motors. After this, the software waits until the next periodic timer interrupt.

3) Industrial Design: We have used a commercial rider glove to make the tactile glove. The glove has two velcro straps, one on the upper arm and another on the forearm. We have attached the main processor board, IMU boards and ERM motors at appropriate locations on the glove. We show the tactile glove in Figure 4. We also show positions of the IMU sensors and tactile feedback actuators in this figure.

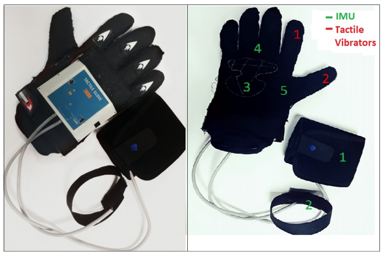

Fig. 4: Positioning of IMUs and actuators in the tactile glove

B. Network Design

The operator end and the teleoperator end communicate over a wired LAN. Here, the operator end consists of the tactile glove and a PC for LAN interface. The tactile glove communicates with PC over serial UART protocol at 230.4Kbps. Similarly, the teleoperator end has a robotic arm and a PC. The robotic arm communicates with the PC again over serial UART protocol at 230.4Kbps. The two PCs communicate over LAN. We mount a USB 2.0 camera on the robotic arm, which captures the scene of the pincher. The camera is connected to the PC at the teleoperator end. The visuals are streamed over the LAN network to the PC at the operator end for display.

C. Teleoperator Controller Design

This includes hardware and software design of the controller board at the teleoperator end.

1) Hardware Design: Hardware design of the controller board includes the design of the processor board at teleoperator end. The processor board houses the central controller (PSoC 5LP), a voltage regulator section, a force sensor circuit and an optocoupler circuit. We use the optocoupler circuit for isolating robotic arm servos and PSoC. Apart from the above, we also have the following modules in the controller board.

a) *Half Duplex UART Clocked at 1Mbps:* The PhantomX reactor robotic arm is built using mechanical links and rotary joints; each of the rotary joints houses a Dynamixel AX-12A servo motor [22]. A servo motor, upon receiving the command, sets the position of the corresponding joint to the desired value. Each servo motor is pre-assigned with a unique address which is used for sending commands. The commands are sent as data packets over UART clocked at 1 Mbps. For this, a half duplex UART module is configured inside PSoC. The UART pins of all the servo motors are daisy chained, and so, all the commands are seen by all the servo motors. However, only the servo motor whose address matches the address in the UART data packet will respond to that command. Dynamixel AX-12A servo motors support sending UART data packets in unicast and broadcast modes. In the unicast mode, data packets intended for only one servo motor are contained in a UART transaction. However, in the broadcast mode, a single UART transaction contains data packets intended for all the daisy-chained servo motors. In the broadcast mode, addresses of the servo motors are included in the corresponding data packets for identification. In our work, we use the broadcast mode for sending the data packets. This is done for two reasons: first, to synchronize movement of all the rotary joints so as to faithfully reproduce operator actions and second, to speed up the response. By choosing the broadcast mode for UART communication over the unicast mode, we increase the speed of command communication by about 50%.

b) *Force Sensing Circuit:* The piezoresistive force sensor is wired across a current source and ground. The circuit converts the sensor resistance to an equivalent voltage. We measure this voltage using a 12bit 100KHz SAR ADC.

2) Software design:

a) *Software Flow:* To start with, we power up the central controller to initialize the PSoC peripherals and also to move the robotic arm to its initial position. Once done with the initialization routine, the software enters an infinite while loop. Every 10 ms, a timer inside PSoC generates an interrupt. At each such epoch, pressure sensor values are read and digitized. These measurements are sent to a PC through a UART link. Also, the robotic arm position information is received from the PC. Using the inverse kinematic algorithm, the position information is converted to link angles which are then used to actuate the corresponding robotic arm links.

b) *Handling robot out of reach condition:* We realize that the link lengths of the PhantomX robot are smaller than what is needed to emulate the movement of a typical human hand. We address this problem by scaling down such legitimate but "out of reach" X, Y, Z coordinates of the hand before feeding them to our inverse kinematics engine. The corresponding joint angles allow the robot's gripper to go close to the desired position. Out of reach condition for the PhantomX reactor robot is given by (see Figure 3)

$$(Z - l_1)^2 + X^2 + Y^2 > (l_2 + l_3)^2$$

We obtain the scaling down parameter k via solving the following equation.

$$\left(\frac{Z}{k} - l_1\right)^2 + \left(\frac{X}{k}\right)^2 + \left(\frac{Y}{k}\right)^2 = (l_2 + l_3)^2 - 1.$$

3) Visual Feedback: We mount a USB 2.0 gooseneck camera on the robot to provide a visual feedback to the operator side.

IV. Results and Discussion

We have made a range of measurements on the system that we have developed. Here we summarize the results and discussion on the same.

A. Tactile Glove

1) Battery Life: We have observed that the battery life of our glove is greater than 3 hours, which is comparable to readily available commercial gloves e.g., VMG 30 Plus (see Table I).

2) Noise: We measured the noise in the estimated (a) wrist position, (b) palm orientation and (c) finger posture. The measurements were done by keeping the glove stationary and collecting 3000 samples. We found the peak to peak noise in x, y, and z coordinates of the wrist to be 0.3mm, 0.63mm and 3.45mm, respectively. We measured the noise in palm orientation and finger posture to be 1.6degree and 0.31degree, respectively. We can see that noise magnitudes are very small.

3) Tracking Accuracy: We next evaluated our wrist position tracking algorithm. The arm length of the operator performing the experiment was measured to be 56cm. We used the same in the program running in glove for this measurement. We asked the operator to stretch his arm along each of X, Y and Z axes of the calibrated global frame. The wrist position estimates given by the algorithm were found to be 55.91cm, 54.87cm and 55.45cm, respectively, which are only marginally different from 56cm.

Next, we asked the person to move his wrist along each of X, Y and Z axes of the calibrated global frame by 15cm. We measured the change in the output of the tracking algorithm to be 11.91cm, 12cm and 15.03cm, respectively, along with the X, Y, and Z axes. Note that accuracy along X and Y axes is not as good as desired because it depends on how the operator is wearing the glove which affects the orientation of the IMU sensors. However, in our setup, the lack of tracking accuracy is not critical since the operator is also assisted with the visual feedback of what is happening at the teleoperator end and he can control the robotic arm appropriately.

We also measured the accuracy of the tracking algorithm in estimating palm orientation. For this, we asked the operator to first keep his palm straight and then perpendicular to his forearm, i.e., at 0degree and 90degree, respectively. We found the palm orientation as given by the tracking algorithm to be 0.1degree and 86.1degree, respectively, in the two cases.

B. Robotic Arm

1) Tracking Accuracy: We also measured the accuracy of the robotic arm in tracking the wrist movement. For this, we commanded the robotic arm to move along each of the X, Y and Z axes by 5cm. The measured movement of the robotic arm was 5.2cm, 4.9cm and 5.1cm along the X, Y and Z axes, respectively.

C. Round trip delay and Component Latencies

We have measured latencies of various components in our system and also the round-trip latency. We show the measured values in Figure 5.

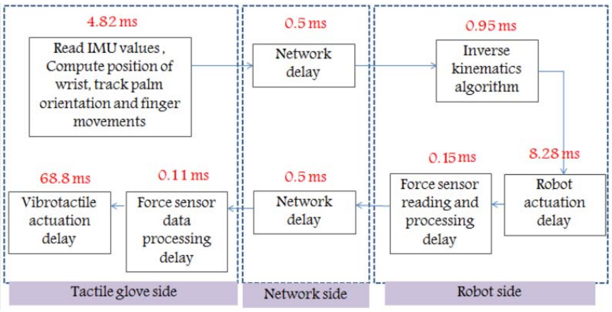
4.82 ms
Read IMU values, Compute position of wrist, track palm orientation and finger movements
0.5 ms
Network delay
0.95 ms
Inverse kinematics algorithm
68.8 ms
Vibrotactile actuation delay
0.11 ms
Force sensor data processing delay
0.5 ms
Network delay
0.15 ms
Force sensor reading and processing delay
8.28 ms
Robot actuation delay
Tactile glove side
Network side
Robot side

Fig. 5: Component-wise latencies. Round trip latency is 85ms. There may be further error of 0-10ms because of periodic sampling of IMU data

Now, we elaborate on each block in Figure 5. We describe our measurement method and also discuss how to improve latencies.

1) Reading IMU values, computing wrist position and tracking palm orientation and finger movements:

a) *Description:* The latency here can be divided into two parts:
   i) Latency involved in reading data sequentially from five 9-axis IMU sensors through the I2C bus at 400Kbits/s. This takes 2ms.
   ii) Latency involved in computing DCMs plus the latency involved in estimating wrist position, palm orientation and finger movement. This amounts to 2.82ms. This is purely a firmware delay.

b) *Measurement Method:* This latency is measured by using a timer component in PSoC. The timer is configured to start counting when the code enters the block. The timer is stopped at the block exit point. The timer value is then used to determine the latency.

c) *Discussion:*

a) Delay in reading data from the IMU sensors can be improved in the following ways.

i) Parallel reading of IMU sensors using multiple I2C master blocks.

ii) Using IMU sensors and master controllers with support for I2C bus speeds greater than 400Kbits/s or using embedded bus protocols supporting higher data rates, e.g., SPI instead of I2C.

b) Improving firmware delay
PSoC Controller used in this work is configured to work at a clock frequency of 24MHz. PSoC can support CPU clock speeds up to 67MHz, so the delay can be cut down by more than 50% by increasing the CPU clock alone. Replacing PSoC with a fast controller like Raspberry Pi can also reduce the firmware delay.

2) Network Delay:

a) *Description:* By this we mean the network delay between the PCs at the operator and the teleoperator ends. This delay is observed to be 0.5ms.

b) *Measurement Method* Average "ping" delay between the two PCs is used to determine this value.

c) *Discussion:* This delay is higher than what is expected in a point-to-point link between two nodes in a LAN, the reason being that our experimental setup is not isolated from external traffic. We propose to use an isolated experimental setup to get the least possible network delay between the two nodes. We can also incorporate a controlled emulated network environment to see the effect of realistic network environments on TCPS applications.

3) Inverse Kinematic Algorithm:

a) *Description:* This is the time taken to execute the inverse kinematic engine coded in PSoC controller. This is a pure firmware delay and is measured to be 0.95ms.

b) *Measurement Method:* Identical to the method described in Section IV-C1.

c) *Discussion:* We can improve this following the same suggestions as made in Section IV-C1.

4) Robot Actuation Delay:

a) *Description:* This is the time taken by the robotic arm to execute an action specified by PSoC controller. We start counting this after the inverse kinematic engine produces an output. This delay is found to be 8.28ms.

b) *Measurement Method:* A human operator wearing the glove is asked to make typical palm movements. The data from the glove is logged every 10ms. From the logged data, the maximum degree change in the palm between two consecutive data samples are identified. In our case, the maximum change was found to be 2.76degree. Then, from the datasheet of the servo motor Dynamixel AX-12A, the amount of time for the motor to rotate 2.76degree is computed. This value turned out to be 8.28ms.

c) *Discussion:* The measurement method we adopted considered the movement of only one robotic joint. However, in reality, when the robotic arm replicates operator's action, multiple joint movements are registered. Thus the response time we have estimated may not be real, nevertheless, it gives a first cut estimate on the performance of the motors used in the robotic arm. With regard to reducing the robot actuation delay, replacing the Dynamixel AX-12A motors in the robotic arm with a fast acting motor is an option.

5) Force sensor reading and processing delay:

a) *Description:* This latency is measured to be 0.15ms which consist of the following two components.

i) Latency involved in reading the flexi force sensor. The primary contributor to the delay here is the ADC used in digitizing the resistance. We use a 12bit SAR ADC that generates 1 sample every 10us.

ii) Latency involved in transferring data from PSoC to the PC (at the teleoperator end) through UART which is configured for a baud rate of 230.4Kbps.

b) *Measurement Method:* Identical to the method described in Section IV-C1.

c) *Discussion:* Latency can be reduced either by increasing the UART baud rate or by replacing the UART bus with SPI or USB interfaces supporting higher speeds.

6) Force sensor data processing delay:

a) *Description:* This is primarily the latency involved in transferring data from PC to the tactile glove through UART configured for a baud rate of 230.4kbps.

b) *Measurement Method:* Identical to the method described in Section IV-C1.

c) *Discussion:* This latency can be reduced either by increasing the UART baud rate or by replacing the UART bus with SPI or USB interfaces supporting higher speeds.

7) Vibrotactile actuation delay:

a) *Description:* This is the actuation delay of the vibrotactile actuators in the glove. We use vibrotactile actuators to provide haptic feedback.

b) *Measurement Method:* A step signal is applied to the vibrotactile actuator from PSoC. The resulting vibrations are then sensed using an accelerometer attached to the actuator. The time constant of the vibrotactile actuator step response is graphically determined from the accelerometer readings.

c) *Discussion:* Alternatives to vibrotactile actuators need to be identified to bring down the haptic actuation delay. More work is required in this area.

V. Conclusion

We, in this work, have demonstrated what a typical control Internet would look like. As a part of this exercise, we have also developed a low-cost tactile glove adequate for executing the task of remotely picking and placing an object. A comparison of our tactile glove with respect to commercially available units is shown in Table I [10], [11]. We also have successfully demonstrated the system in which the two ends of the TCPS, namely the operator and the teleoperator ends, are connected over LAN.

TABLE I: Feature comparison of tactile gloves

Glove	Our Work	CyberTouch II	VMG 30 Plus
Cost (in $)	160	36285	7000
Tactile feedback	Yes. 2 No's.	Yes. 6 No's.	Yes. 5 No's.
Tracking Type	Wrist, Palm, 2 Fingers	All Fingers.	Wrist, Palm, All Fingers.
Update Rate	100 records/sec	90 records/sec	90 records/sec
Interface	UART	UART	UART
Battery Backup	3 Hours	-	4 Hours

Acknowledgment

We thank RBCCPS at IISc, Bangalore and INSPIRE Faculty Research Grant (DSTO-1363) for partially supporting this work. We also thank Prof Fernando A. Kuipers and Belma Turkovic from the TU Delft, The Netherlands for their support.

References

[1] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5g-enabled tactile internet," *IEEE Journal on selected areas in communications*, vol. 34, no. 3, pp. 460 – 473, March 2016.

[2] G. Fettweis, "The tactile internet: Applications and challenges," *IEEE Veh. Technol. Mag*, vol. 9, no. 1, pp. 64–70, March 2014.

[3] M. Dohler, "The tactile internet iot, 5g and cloud on steroids," in *5G Radio Technology Seminar. Exploring Technical Challenges in the Emerging 5G Ecosystem*. IEEE, March 2015, pp. 16–16.

[4] A. Rodriguez-Angeles, A. Morales-Diaz, J. C. Bernab, and G. Arechavaleta, "An online inertial sensor-guided motion control for tracking human arm movements by robots," in *2010 3rd IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, Sept 2010, pp. 319–324.

[5] I. Prayudi and D. Kim, "Design and implementation of IMU-based human arm motion capture system," in *2012 IEEE International Conference on Mechatronics and Automation*, Aug 2012, pp. 670–675.

[6] F. F. Mohammadzadeh, S. Liu, K. A. Bond, and C. S. Nam, "Feasibility of a wearable, sensor-based motion tracking system," *Procedia Manufacturing*, vol. 3, pp. 192 – 199, 2015, 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2351978915001298

[7] H. Zhou and H. Hu, "Kinematic model aided inertial motion tracking of human upper limb," in *2005 IEEE International Conference on Information Acquisition*, June 2005, pp. 6 pp.–.

[8] P. Ben-Tzvi and Z. Ma, "Sensing and force-feedback exoskeleton (safe) robotic glove," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 6, pp. 992–1002, Nov 2015.

[9] P. Weber, E. Rueckert, R. Calandra, J. Peters, and P. Beckerle, "A low-cost sensor glove with vibrotactile feedback and multiple finger joint and hand motion sensing for human-robot interaction," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Aug 2016, pp. 99–104.

[10] (2017) Cybertouch ii, cyberglove systems. [Online]. Available: http://www.cyberglovesystems.com/cybertouch2/

[11] (2017) Vmg 30 plus, virtual motion labs. [Online]. Available: http://www.virtualmotionlabs.com/vr-gloves/vmg-30-plus/

[12] G. A. D. Anjos, G. A. Santos, J. A. D. Amado, and J. E. S. Marques, "Haptic glove development for passive control of a robotic arm," in *2016 IEEE INTERCON*, Aug 2016, pp. 1–6.

[13] M. Yaqoob, S. R. Qaisrani, M. Waqas, Y. Ayaz, S. Iqbal, and S. Nisar, "Control of robotic arm manipulator with haptic feedback using programmable system on chip," in *2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE)*, April 2014, pp. 300–305.

[14] (2018) Website of kurian polachan. [Online]. Available: https://sites.google.com/view/kurianpolachan/tcps-research

[15] (2017) Invensense, mpu-9250. [Online]. Available: https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/

[16] W. Premerlani and P. Bizard, "Direction cosine matrix imu: Theory," 01 2009.

[17] A. Bose, S. Puri, and P. Banerjee, *Modern Inertial Sensors and Systems*. Delhi, India: Prentice Hall India, 2009.

[18] (2011) Starlino, dcm tutorial- an introduction to orientation kinematics. [Online]. Available: http://www.starlino.com/dcm_tutorial.html

[19] (2017) Vibrating mini motor disc, adafruit. [Online]. Available: https://www.adafruit.com/product/1201

[20] (2017) Psoc, cypress semiconductors. [Online]. Available: http://www.cypress.com/products/32-bit-arm-cortex-m3-psoc-5lp

[21] (2017) Battery technologies, sparkfun. [Online]. Available: https://learn.sparkfun.com/tutorials/battery-technologies

[22] (2017) Ax-12a, crustcrawler robotics. [Online]. Available: http://www.crustcrawler.com/products/bioloid/docs/AX-12.pdf

[23] (2017) Phantomx reactor, trossen robotics. [Online]. Available: http://learn.trossenrobotics.com/interbotix/robot-arms/reactor-arm.html

[24] tekscan. (2017) Flexiforce sensor. [Online]. Available: https://www.tekscan.com/product-group/embedded- sensing/force-sensors