DeFatigue: Online non-intrusive fatigue detection by a robot co-worker

Pradip Pramanick and Chayan Sarkar
Embedded Systems and Robotics
TCS Research and Innovation, Kolkata, India
{pradip.pramanick, sarkar.chayan}@tcs.com

***Abstract*— A robot as a companion or co-worker is not an emerging concept anymore, but a reality. However, one of the major barriers to this realization is the seamless interaction with the robots that includes both explicit and implicit interaction. In this work, we assume a use-case where a human and a robot together carry a heavy object in a co-habitat (home or workplace/factory). Two human beings while doing such a work understands each other without explicit (vocal) interaction. To realize such behavior, the robot must understand the fatigue state of the human co-worker to enable seamless work experience and ensure safety. In this article, we present *DeFatigue*, a non-intrusive fatigue state detection mechanism. We assume that the robot's hand is equipped with a force sensor. Based on the change of force from the human side while carrying the object, *DeFatigue* is able to determine the fatigue state without instrumenting the human being with an additional sensor (internally or externally). Moreover, it detects the fatigues state on-the-fly (online) as well as it does not require any (user-specific) training. Based on our experiments with 18 test subjects, fatigue state detection by *DeFatigue* overlaps with the ground truth for 85.18% of the cases whereas it deviates 4.09 s (on average) for the remaining cases.**

I. Introduction

Industrial revolution paved the way for using robots inside various industry and assembly-line. Though the industrial robots are becoming smarter in terms of performing more delicate tasks, their usage is mostly limited where it is unproductive or unsafe to employ human beings. Nowadays, we see robots in different roles in our daily surroundings ranges from tutor, guide, assistant, co-worker, etc. In this work, we focus on a situation where a robot and a human work as a co-worker to jointly carry a heavy object from one place to another. According to a survey by Ray *et al.* [15], moving heavy objects is among the top five tasks that people expect from a household robot. A recent study also showed that people collaborate best with a proactive robot that can understand when to help a human [3]. Our goal is to enable a worker robot to detect localized muscle fatigue of a human co-worker while jointly carrying a heavy object. This way the robot can act accordingly as the human co-worker gets fatigued without explicit (vocal) communication. Also, the human being herself may not be able to assess the muscle fatigue state accurately. Thus, implicit fatigue detection will ensure a safe and seamless interaction between a robot and a human in a co-working environment be it home, workplace or industrial setup.

Challenges. The goal of identifying localized muscle fatigue is to avoid any form of injury by taking proactive action and provide additional support to the human co-worker if possible. Since muscle fatigue can happen within a short time frame and the consequence can be severe, it is required to identify on-the-go (online). As the time to get fatigue varies significantly from person-to-person, the detection should be people agnostic as well. Otherwise, the system needs to be aware of each individual's capability and physique. This involves user-specific training and adaptation. Moreover, the detection needs to be non-intrusive to be usable in practical situations like work environment. There exists some effort that can identify muscle fatigue, but they require the human subject to wear some sensors [4], [9], [17]. Such a setup is neither cost effective nor feasible. On the other hand, there are video-based non-intrusive techniques that can identify overall fatigue state and mostly utilize the drowsiness of the subject [12], [18]. However, the applicability of these techniques to identify localized muscle fatigue is limited.

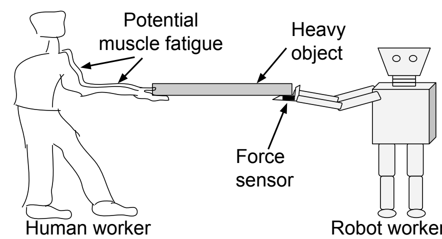


Fig. 1: Schematic diagram of *DeFatigue*.

Approach. In this article, we develop *DeFatigue*, a non-intrusive mechanism that detects fatigue state of a human co-worker during the working state without any explicit interaction. As most of the robotic hands are equipped with a force sensor(s), *DeFatigue* leverages on it to identify the muscle fatigue state. A schematic diagram is shown in Fig. 1. In a nutshell, it observes the change of force from the human side, which is reflected in the force sensor at the robot side. If force change is significant enough, it can be identified as a change of fatigue state. However, mere force change based classification can lead to a false positive, i.e., detection of a fatigue state even if it is a small perturbation or hand tremor. To avoid these situations, a threshold based approach can be applied. However, a strict threshold can lead to miss-detection, which may lead to an accident. Moreover, the change of force is highly dependent on the physique of the person. To overcome all these limitations, *DeFatigue* uses a moving window based average force change and compare it with initially applied force. This provides a generic approach and mitigates the person-to-person variations.

TABLE I: Muscle fatigue is classified into four states with associated discomfort and tremor. [8]

Muscle Performance	State I	State II	State III	State IV
muscle discomfort	tightness or slight cramping	continuous cramping with intermittent pain	continuous pain and desire to abandon	unable to sustain activity
hand tremor increase	138%	225%	300%	350%

Contributions. We experimented with 18 human subjects and a robotic hand is emulated by a force sensor mounted on a table corner. *DeFatigue* is able to detect the fatigue states with very high accuracy when compared to the ground truth (based on feedback from the human being during the data collection). Specifically, our contributions are as follows.

- We develop an online, multi-stage muscle fatigue detection mechanism that can be utilized by a robot co-worker while jointly carrying a heavy object.
- The proposed method is non-intrusive and user agnostic. Thus, it does not require the user to wear any sensor (internally or externally) and it does not require any user-specific training or adaptation.
- *DeFatigue* shows a very small number of miss/over-detection across all the test cases. Moreover, *DeFatigue* shows significantly less number of false negatives as compared to false positives. This ensures the accuracy of the system along with safety in the work environment.

II. Related Work

In this section, we discuss some of the existing work on fatigue detection. As muscle fatigue is not an instantaneous event, we first discuss how to quantify fatigue states.

A. Fatigue quantification

The decision about fatigue is not a binary decision, especially the muscle fatigue. Rather, it is a continuous accumulation of lactic acid while the muscle is under stress. When two human beings collaborate, they often understand the fatigue state (may not be quantitative) of the other person without explicit communication and act accordingly. Now, if one of the co-workers is a robot, it is essential to assess the fatigue state with every passing moment and decide whether to persist on the job or not. In this work, we are interested in detecting fatigue state of a person of unknown physical strength, which varies significantly from person to person. In order to be able to detect fatigue state of a person, it needs to be quantified first. Though such a quantification should be people agnostic, the time to reach muscle fatigue generally depends upon several physiological properties of the human body such as age, height, weight, gender, physical wellness, etc. [6], often cumulatively termed as physical strength. Moreover, there is also temporal variations of physical strength for the same person, which makes the detection problem more challenging.

Don B. Chaffin [8] defines localized muscle fatigue of hand with respect to physiological properties of the human body. Instead of a binary fatigue and non-fatigue classification, he defines a coarse-grained classification of four states of fatigue (Table I). Though these four fatigue states are described from a human perception point of view, by correctly classifying these states, a robot can ensure safety and provide a more accurate amount of support. Depending upon the specific requirement, any of the four states can be chosen as the final alert state to trade off safety with task efficiency.

B. Fatigue detection

In the literature, the most popular and reliable non-invasive methods of detecting localized muscle fatigue are based on surface electromyography(sEMG) [1], [4]. Though sEMG is declared as non-invasive, it is not a truly non-intrusive sensing technique because it requires sensors to be physically attached to the human body and thus defeats the purpose of natural human-robot collaboration. Halim *et al.* [11] did a study on analyzing muscle fatigue using mean power frequency of sEMG in a manual lifting task of variable heights. The study shows that out of a set of different muscles (*Biceps Brachii* and *Erector Spinae* of left and right hands) only a subset can get fatigued depending upon different lifting heights and pose. So it can be argued that for a robot to detect fatigue state of a human co-worker, all the muscles involved in the work need to be attached with sensors leading to even more invasiveness. Spyropoulos *et al.* [16] also did a similar experiment of repetitive lifting task and assessed fatigue of upper limb involving sEMG. Brown *Brown et al.* [7] used 3D accelerometer attached to *Biceps Brachii* along with sEMG to detect fatigue while subjects did a set of repetitive exercises. In a very recent work by Peternel *et al.* [14], wireless sEMG sensors were used to detect muscle fatigue while performing collaborative tasks with a robot. However, their system is able to detect only binary fatigue states and it requires use-specific calibration. Also, because of a binary detection, the robot only starts to adapt after the fatigue state has been reached. Whereas a multi-stage fatigue detection can help the robot adapt more pro-actively.

As an alternative, force sensors can be considered as a fatigue measurement device which has been previously used for drivers to assess their drowsiness and overall fatigue level [10], [13]. Image and video processing techniques with machine learning are also very popular and accurate methodologies to assess driver's fatigue level [12], [18]. However, we argue the image and video processing methods are less applicable to detect localized muscle fatigue. On the other hand, force sensor based techniques can be more suitable for such a scenario since the data produced by a forced sensor provides a more localized assessment of a human body.

Chieh *et al.* [10] describe a methodology that employs two force sensors on a steering wheel to collect driver's grip force data during simulated driving. The authors conclude that fatigue detection is a complex matter and should not be done by merely checking the threshold. This is because of the fact that due to variations of physical strength, the change in force during the transition to fatigue state will vary, making the determination of the threshold infeasible. The collected data is processed off-line to estimate the mean grip force during alert state and fatigue state. With the estimated mean values, the authors claim to detect the point of fatigue using mean change detection algorithm called *CUSUM*, which is a well-known statistical change-point detection algorithm. However, the described method in [10] is offline as it requires the estimated mean force corresponding to fatigue state to be known a priori. Also, the authors haven't discussed the performance of the algorithm in terms of accuracy for multiple subjects. Lee *et al.* [13] use an FSR-408 strip sensor on the steering wheel to collect grip force data of a driver during a simulated driving task to measure the subjective sleepiness of the driver. They have taken an average dataset from 31 participant's data and used a statistical change detection method called *Repeated Measures ANOVA* to report a single fatigue state. However, they have not discussed how the data varies with the participants. They also describe offline detection and how to use *ANOVA* in an online setting is not discussed. Bhardwaj *et al.* [5] conduct a similar experiment with commercial tactile gloves which contains multiple force sensors. Though the article deals with driver's discomfort and not directly with fatigue, still it shows that force sensors can effectively capture the change in driver's physical properties related to fatigue.

To the best of our knowledge, none of the existing methods that use force sensor is suitable for a robot co-worker to detect multi-state localized muscle fatigue on-the-go. Though the most popular methods use sEMG signals [2], [7], [14], they require electrodes to be put on the person. Clearly, such an invasive sensing method has very few use cases and certainly won't enable natural and implicit communication in a human-robot co-working scenario.

III. System requirements

If a person gets fatigued, continuing any exhaustive work, such as carrying a heavy object can lead to an accident. When a person uses only a body part, such as hand or leg to do some work, lactic acid gets accumulated in the muscle and leads to localized muscle fatigue. In such cases also, continuing the work may lead to an accident, even though the person may not be fully exhausted. The goal of this work is not to identify the overall fatigue state of a human being, but to determine the localized muscle fatigue state.

The idea is that a worker robot detects the fatigue state of a human co-worker while jointly doing a work such that precautions can be taken to ensure the safety. The proactive action by the robot without any explicit communication from the human not only avoids an accident, this would imitate implicit communication between the human and the robot.

The goal of *DeFatigue* is not to achieve a binary fatigue classification system as this may lead to a lower productivity (in case of permissive detection) or higher accident probability (in case of very strict detection). As a result, *DeFatigue* classifies four fatigue state as described in Table I, which is a better quantitative measure. Characterization of these states can be used to select the "dangerous" fatigue state depending upon the safety requirements of a particular use case.

In the observations related to driver's fatigue detection, the exerted force continue to decrease as the person gets tired; ultimately reaching the minimum level, which resembles fatigue state. In our experiments, we have also seen somewhat similar force output. However, minimum force output may not be sufficient to decide multiple fatigue states. Besides, due to variations in the data for different persons, it is not feasible to determine absolute amount of change in force for different fatigue states. Moreover, the presence of high level of noise due to twitching and tremor of the hand, which may not be present in the driver's grip force scenario, make the statistical significance or change detection methods infeasible.

Based on these observations, we enlist some of the requirements for the proposed fatigue detection system.

- **Online detection:** The detection system cannot wait for a stream of future data from the sensor to establish the current fatigue state. Even though analyzing the whole time-series data may accurately classify the fatigue state change in the time domain, it might be too late to avert the possibility of an accident. As the goal of the system is to provide timely detection of fatigue state such that proactive actions can be taken to avoid any accident, online detection is a must requirement for such a system.
- **Simplicity:** Since the system needs to detect the current fatigue state in a time-bounded manner, the sensor data processing should be performed locally. As a result, the algorithm should be simple enough to run on-board, even for a low-cost robotic system, and provide output in semi-real-time. If we choose a complex algorithm that needs to be offloaded to run on a server, reliability may decrease due to network latency and disconnection.
- **Adaptivity and robustness:** There is a significant variation of physical strength among different persons. Moreover, the system should not assume that the human worker would always hold the object in a certain way. Rather, it should be agnostic to the holding pattern of an individual and the associated noise induced due to the holding pattern. Instead of customizing the algorithm for each person, it should be adaptive for any human co-worker, irrespective of her physical strength, holding pattern, etc.
- **Non-intrusive:** The proposed system should not depend on a body-attached sensor on the human being. It not only causes discomfort, it totally defeats the purpose of imitating the implicit communication. Moreover, once a human co-worker gets fatigue and the task is yet to complete, a second human co-worker may replace her. But it becomes very difficult to quickly replace all the sensors for the second co-worker. Thus, the data collection should be non-intrusive.

IV. ONLINE FATIGUE DETECTION

Modeling muscle fatigue is difficult since it can vary significantly from person-to-person. It depends on a person's physical ability, which is related to age, gender, BMI, etc. Fig. 2 shows the response time of different person when they lift a heavy object (same weight) and requested to indicate when they feel it a bit heavier than the initial weight (state 1 fatigue).

Even though there are four states of fatigue, considering the safety of our test subjects, we conducted experiments only until the participant reaches the third fatigue state. Analyzing the collected data and the corresponding ground truth, we found that, to effectively detect and classify three states of fatigue with very high accuracy, some assumptions must be made. These assumptions are based on the observations of the data collection experiments.

1) Let F_I be the average force output of the sensor during an initial time window T_I. We assume that during this time window, the human being is in the non-fatigue state. Thus, F_I reflects the force exerted by the hand muscle during the non-fatigue state. To avoid the effect of noise and outliers, a robust statistic *viz. median* is used to calculate F_I. Let D be a set of n sensor samples collected in T_I. Reordering D such that $d_1 < d_2 < d_3 \ldots d_{n-1} < d_n$, then F_I is given by

$$F_I = \begin{cases} d_{(n+1)/2}, & \text{when n is odd} \\ \frac{d_{(n/2)}+d_{(1+(n/2))}}{2}, & \text{when n is even.} \end{cases} \quad (1)$$

2) Let δT denote the time to reach the first fatigue state, i.e., the time difference between the start of the task and having the first state of fatigue. Then, the next state of fatigue is likely to occur after $\delta T/2$ time unit. This assumption is based on the fact that during the experiments, the next state of fatigue happens roughly in the following δT time unit, as shown in Fig. 2, but never before $\delta T/2$.
3) The transition from non-fatigue to fatigue state is not instantaneous, rather it happens during a short period of time. So a time window is required to analyze the trend of the sensor data for online detection. Let F_c be the current mean force observed during time window T_c. Then the normalized decrease in force from the initial average force (F_I) is denoted as,

$$\delta F = \frac{F_I - F_c}{F_I} \times 100\%.$$

It is assumed that a normalized decrease in force by more than ($\delta F >$) 10% indicates a significant change; thus indicating the chance of "potential" fatigue. This δF is the cut-off threshold to find the window of interest in the force sensor data that is subject to further analysis. We chose this value to reduce false positives. This threshold value is appropriate since all the marked fatigue states from our dataset have more than 10% decrease in force. However, this threshold can be modified to trade off safety with accuracy.

Based on the above-mentioned assumptions, the algorithm for online fatigue detection is designed as shown in Fig. 3. The algorithm starts at the beginning of a human-robot collaborative task, which is triggered by a non-zero sensor reading. The algorithm terminates when the sensor reading is zero, indicating end/abortion of the task. The pseudo-code of the algorithm is described in the following.

Algorithm 1: *DeFatigue*: Algorithm for online non-intrusive fatigue detection.

```
Read F_I using Eq. 1;
cut_off = 10;
while (sensor_reading > 0) do
    Select time window T_c;
    while (!T_c.full()) do
        Fill T_c with force sensor reading;
    Calculate δF for time window T_c;
    if (δF > cut_off) then
        Mark starting of potential fatigue;
        Select time Window of Interest, T_WOI;
        While T_WOI is not full, fill T_WOI with δF;
        if Slope(T_WOI) > 0) then
            Mark next fatigue state;
        Calculate δT;
        Wait for δT/2 time;
    Undo marked potential fatigue;
```

Algorithm 2: Algorithm for finding slope of T_{WOI}.

Let δF_1,$\delta F_2 \ldots \delta F_n$ denote the data points in T_{WOI};
Get the midpoint of T_{WOI} as δF_m;
Calculate M_1 as $M_1 = \frac{\sum_{i=0}^{m-1} \delta F_i}{m}$;
Calculate M_2 as $M_2 = \frac{\sum_{i=m}^{n} \delta F_i}{m}$;
Calculate δM as $(M_2 - M_1)$;
return δM

When the normalized force change (δF) is more than the cut-off threshold, a window of interest (*WOI*) opens up. The slope of the force data within this *WOI* determines whether a fatigue state transition has occurred or not. A decreasing trend in force during the *WOI* provides a clear indication of fatigue. If the data-points inside *WOI* is represented as a line, then a positive slope in the line denotes a decreasing trend in force. This is because *WOI* is filled with the normalized change *dF*, which becomes positive when force is decreasing. However, to make the algorithm suitable to run on an embedded system like Arduino, the slope of the line is determined by a computationally simple method, described below.

V. EXPERIMENTAL SETUP

To conduct the experiment of fatigue detection while maintaining safety and without the burden of additional

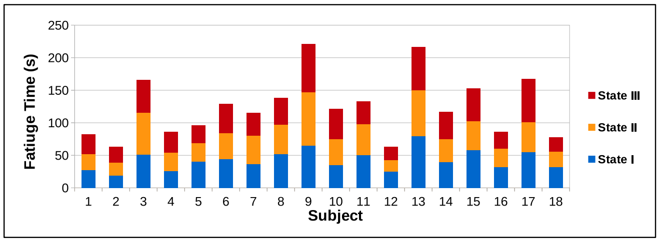


Fig. 2: Variations in time across the test subjects to reach up to the 3rd fatigue state.

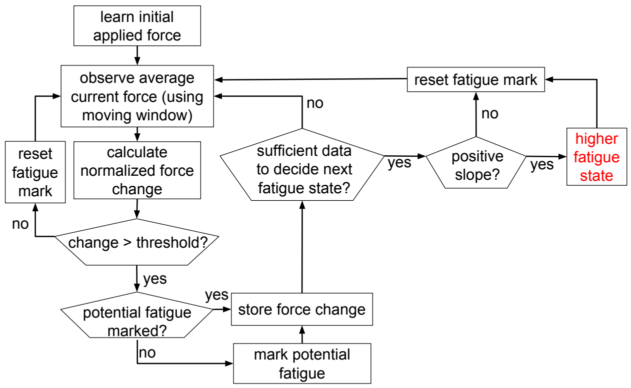
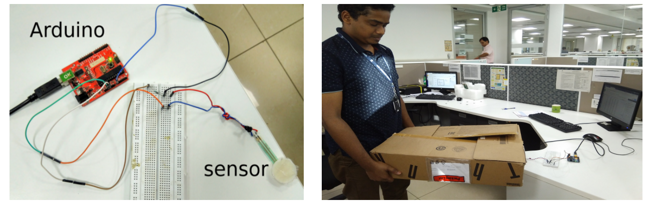


Fig. 3: Flow diagram of the online fatigue detection.

cost and complexity, we have resorted to using a simple and realistic technique to model a robot's behavior in our experiment. During the data collection process, a force sensor is mounted on the corner of an oval-shaped, rigid, wooden table and connected to an Arduino board. It is used to simulate a humanoid robot's hand. Putting a box on the sensor simulates a robotic hand holding the box and the sensor on the corner of the table simulates a sensor on the palm of a robotic hand. The weight of a heavy object held by the robot falls on the sensor, which measures the change in force with respect to time. It is assumed that the robot is strong enough to hold the box indefinitely, which closely relates to the properties of a fixed, rigid table.

A. Apparatus

As mentioned earlier, the setup consists of a force sensor. The sensor has a robust polymer thick film (PTF) that exhibit a decrease in resistance with the increase of applied force. A voltage divider is used to convert resistance output to voltage output, which has a larger scale. This means that the voltage output of the sensor is increased with increasing force. The sensor is connected to an Arduino board to collect the force output data along with timestamps. During initial data collection period, the Arduino board is connected to a personal computer to store the data. During online detection, the detection algorithm runs on the Arduino board itself. A heavy rigid box, used as a payload, is held by the participants from one end whereas the other end is rested on the sensor. We use an FSR-402 as the force sensor that has a sensitivity of 0.2 N to 20 N. It has a circular sensing region with a diameter of 12 mm, which is suitable to be put on a humanoid robot's hand. However, beyond the outer ring of the sensing



(a) sensor (b) data collection

Fig. 4: Force sensor and Arduino based experimental setup and data collection with test subjects.

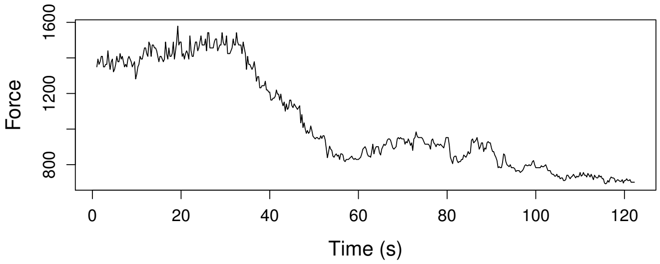
region, there is a protective layer which prevents any object larger than the sensing region to give accurate output. Also, the sensing region resides at a little lower height than the protective layer. So, for a rigid box with a flat base, which is larger than the sensor, the weight does not fall on the sensing region completely. To overcome this, a small, plastic, 3D-printed cylinder with a small base and large top is used as shown in Fig. 4a. Because of this setup, the weight of box falls over the larger flat top of the cylinder. Even though the total weight of the box is not reflected by the sensor output, the change in force on the sensor is well measurable, and that is what we are interested in.

B. Methodology

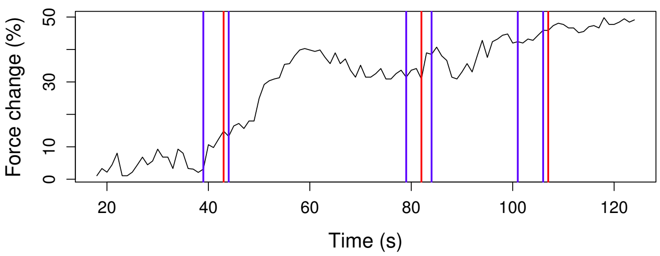
During the initial data collection phase, the participants are asked to hold the box from one end, roughly at the same initial angle, it made with the table as shown in Fig. 4b. Before that, the participants are informed about the fatigue states according to the observations in Table 1. They are requested to make a mental scale of the states and to speak out when he/she is feeling the next state of fatigue. The timestamps are manually marked to build the ground truth data. No visual feedback of the force sensor output was shown to the participants to avoid bias on ground truth. Keeping in mind the safety of the participants, the data collection experiments are conducted until the participants reach state III fatigue only. Participants are selected from a wide age group of 24 to 40 years old. Both male and female participants are experimented with. To introduce further variation in the physical strength of the participants, persons with varying BMI are chosen. Also, multiple experiments are conducted for the same participant at different times to capture the temporal variation of physical strength.

VI. Evaluation

As mentioned earlier, our goal is to detect the fatigue state transition on-the-go such that proactive measures can be taken in real-time. Thus the accuracy of detection along with timeliness is a requirement. So, at first, let us evaluate the accuracy of *DeFatigue*. As our algorithm use certain parameters (as mentioned in Section IV), we also evaluate how these parameters are chosen that withhold the good performance. Fig. 5a and 5b show a sample force sensor output and the corresponding normalized change in force, respectively. Fig. 5b also shows the detected fatigue states by *DeFatigue* as indicated by a window of blue lines. These lines denote the starting and ending points of *WOI* for which the potential fatigue is marked as positive. The vertical red

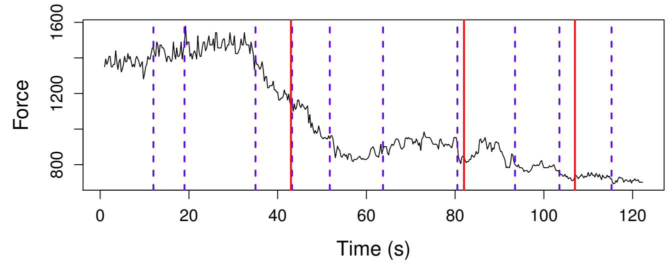


(a) sensor output

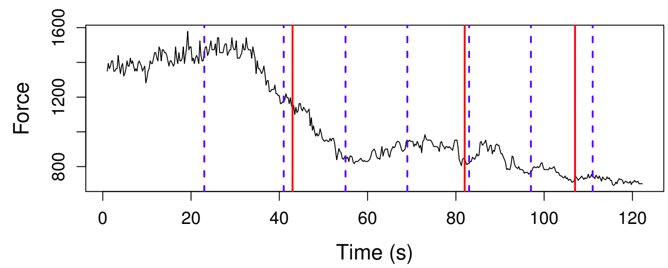


(b) normalized force change

Fig. 5: Detected fatigue indicated by a window (pair of blue lines) and the ground truth (indicated by the red line).



(a) CUSUM



(b) CUSUM with skips

Fig. 6: Sample detection points (dotted blue lines) using Chieh *et al.*'s method and ground truth (red lines).

lines denote the ground truth provided by the subject. It is clear from this figure that the fatigue state transition is accurately indicated by *DeFatigue* as the ground truth falls within (or within close proximity of) the estimated window of transition. We report the results for all the subjects in terms of false positives and false negatives. A false positive is marked when a *WOI* is given positive fatigue by the detection algorithm even though ground truth reported by the subject is not there. On the other hand, when the system misses a state transition, it is marked as false negative. To report the overall accuracy of the algorithm based on the data collected from the 18 subjects, we use the following formula.

$$A = \frac{S - (F_P + F_N)}{S} \times 100\%, \tag{2}$$

where S, F_P, and F_N are the total number of fatigue states, false positives, and false negatives, respectively. For 18 subjects, the total no. of fatigue states to be detected is $18 \times 3 = 54$. So the overall accuracy is

$$A = \frac{54 - (6 + 2)}{54} \times 100\% = 85.18\%.$$

We have compared *DeFatigue* with the methodology described by Chieh *et al.* [10] which used *CUSUM* algorithm to detect significant mean changes in the force sensor data. Even though the described method works offline, we have converted the algorithm to detect changes online. The modified *CUSUM* based change detection algorithm buffers data in a time window that is of the same size of the window used by *DeFatigue*. Then the algorithm analyzes the window and reports if there is a significant change. We observe that the algorithm by Chieh *et al.* is very sensitive to perturbation and thus reports more false positives than *DeFatigue*, as shown in Fig. 6a for the same subject shown in Fig. 5. The poor performance of Chieh *et al.*'s method is attributed to the fact that it is not catered for noisy data. In case of driver's fatigue detection, hand grip force data contains lesser noise and there is high similarity among different subjects. Whereas in robot co-worker scenario, the sensor data incorporates noise from hand movements and varies significantly for different subjects.

Fatigue not being an instantaneous event, there is a chance of falsely detecting the after-effects of a fatigue state transition as another state transition. As seen in Fig. 6, the detected fatigue states by Chieh *et al.*'s algorithm are very close to each other and some of them are actually the after-effects of the previous fatigue state transition. We have already discussed in section IV that to avoid false positives of the after-effects, the algorithm should wait for $\delta T/2$ time unit before checking for next fatigue state. Thus, we introduce the waiting mechanism in Chieh *et al.*'s method and it significantly reduces the rate of false positives. We call it *SKIP_CUSUM*. A sample comparison of the same subject is shown in Fig. 6b. A detailed comparison of the three methods for all the 18 subjects is shown in Fig. 7. Note that due to high sensitivity to mean change, both *CUSUM* and *SKIP_CUSUM* reports zero false negative. However, *CUSUM* reports a large number of false positives. *SKIP_CUSUM* performs better than *CUSUM*, but *DeFatigue* outperforms both in terms of accuracy per subjects. *SKIP_CUSUM* reports a total of 33 false positives for the entire dataset, which results in 38.89% accuracy using Eq. 2.

In our experiment, we have observed that fatigue being both physiological and psychological state, the subject may not report fatigue just as it happened. By visually analyzing the ground truth and the trend in force sensor data, we found that sometimes the subjects reported fatigue a few seconds after it has actually happened. So the reported ground truth may not exactly fall inside the positive *WOI* reported by the algorithm, but sometimes it follows the window closely. So, apart from reporting accuracy in terms of the number of false positives and false negatives, we have also evaluated the accuracy with respect to the time difference between the ground truth and the positive *WOI*. The error of reporting

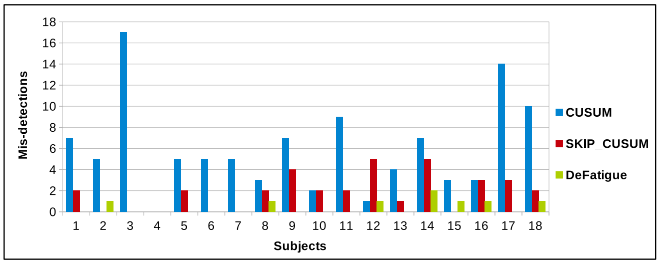


Fig. 7: Fatigue detection using Chieh *et al.*'s method v/s *DeFatigue*.

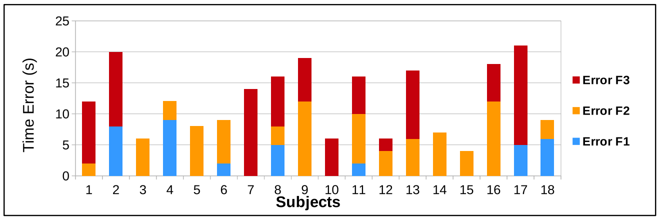


Fig. 8: Detection error with respect to time.

fatigue by the algorithm is shown in Fig. 8. In the figure, "Error F1" signifies the deviation of the detected window of first fatigue state from the ground truth, "Error F2" is for the second fatigue state, and so on. When the ground truth falls within the detected window, the deviation is zero; so the corresponding error is not shown in the figure.

We have selected various parameters like the size of the initial time window (T_I), cut-off threshold, size of *WOI*, etc., based on our observations to maximize safety. These parameters are correlated with the nature of the task, i.e., how long the task completion should take, what is the shape and weight of the jointly manipulated object, etc. However, we assume by collecting a little amount of data for a task, the robot can learn about the correlation and hence adjust the thresholds dynamically. For the results shown in Table II, we use a box weighting 20 kg and select an initial time window of 10 s, cut-off threshold 10%, and size of *WOI* as 5 s.

We evaluate the detection accuracy by varying the parameters as shown in Table II. It is evident that increasing the cut-off threshold while keeping the size of *WOI* fixed, results in lesser number of false positive. This is because, with a larger cut-off threshold, the system becomes lesser sensitive to force change. The system detects a state transition only when there is a significantly larger change in force, which of course results from a fatigue state transition. However, with a larger cut-off threshold, the number of false negative also starts to increase. A larger cut-off means, there is a chance of missing out a fatigued state for physically stronger persons whose force change is lesser. Similarly, increasing the size of *WOI* reduces the number of false positives, as a larger *WOI* clearly captures the decreasing force trend. However, having a large *WOI* means the algorithm has to wait for a long time before giving the alert, which is not desirable. Initially, by collecting data, the robot can produce such a table, and select the row which is optimal for the selected task. The more the robot learns, more accurately it can fine-tune the parameters.

TABLE II: Accuracy of *DeFatigue* for varying parameters.

Cut-off(%)	**WOI(s)**	**False Positives**	**False Negatives**	**Accuracy (%)**
6	5	12	1	77.8
8	5	8	2	81.5
10	5	6	2	85.3
12	5	7	3	81.5
10	3	12	2	74.1
10	7	9	2	79.7
10	9	8	2	81.5
10	12	6	3	83.3

However, merely selecting the row with the highest accuracy may not be suitable for this. For safety considerations, a false negative is more undesirable than a false positive. So a row with the minimal weighted average of false positive and false negative is to be selected, where the weight of false negative is more and can be selected based on the safety requirements. In fact, the traditional concept of false positive may not be applicable here. The online algorithm is designed in such a way that any supposedly false positive w.r.t ground truth is actually marked as an early warning. As seen in a sample detection plot in Fig. 9, there exists a false positive between the 2nd and the 3rd states of fatigue. However, the online algorithm will mark the false positive window as the next fatigue next (3rd fatigue state in this case). By looking at the total data, we can see that marking of false positive leads to an early marking of the 3rd state of fatigue. Even though this reduces the productivity of the setup, it does not compromise the safety of the system.

The time difference between the early-marked fatigue state and the ground truth is another measure of accuracy. We also evaluate *DeFatigue* based on this time deviation measure. It is to be noted that as the time to reach fatigue states varies for different subjects, the accuracy measured by time deviation must be normalized. For example, a smaller deviation may be critical for a subject with relatively quicker to reach fatigue, while the same deviation may not be critical for a subject with much larger time to reach fatigue. So to normalize, we calculate accuracy using the following formula for all the subjects and then take the average as the overall accuracy. Let, F_1, F_2, F_3 denote the time to reach the first, second, and third fatigue states, respectively and Et_1, Et_2, Et_3 denote the respective time deviation by *DeFatigue*. Then timing accuracy for an individual subject is given by,

$$A' = \frac{(F_1 + F_2 + F3) - (Et_1 + Et_2 + Et_3)}{(F_1 + F_2 + F_3)} \times 100\%.$$

Using this formula, we calculate time accuracy for all the 18 subjects as shown in Table III. Based on these accuracy figures, we calculate an average accuracy of all the subjects to be 88.83%. Also, the total time deviation being 221 seconds, time deviation from ground truth per fatigue state is calculated to be 4.09 seconds.

VII. Conclusion

In this article, we develop *DeFatigue*, a non-intrusive, online method to detect multiple fatigue states of a human

TABLE III: Accuracy of *DeFatigue* in terms of deviation (in seconds) of the detection window from the ground truth.

Subject	**Et_1+Et_2+Et_3**	**F_1+F_2+F_3**	**Accuracy (%)**
1	12	82	85.4
2	20	63	68.3
3	6	166	96.4
4	12	86	86.1
5	8	96	91.7
6	9	129	93
7	14	115	88.6
8	16	138	88.4
9	19	221	91.4
10	6	121	95
11	16	133	87.9
12	6	63	90.1
13	17	216	92.1
14	7	117	94
15	4	153	97.3
16	18	86	79
17	21	167	87.4
18	10	76	86.8

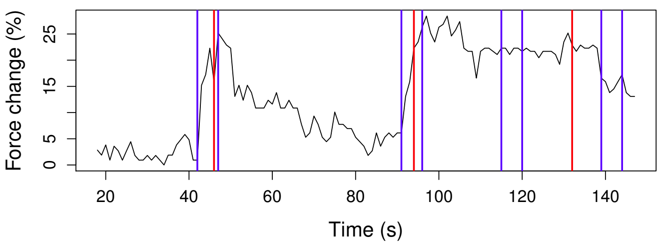


Fig. 9: False positive - the fatigue detection is indicated by a window (marked by the blue lines) and actual fatigue state transition (the ground truth) is indicated by the red line.

worker by a robot co-worker when they jointly carry a heavy object. Non-intrusive refers to no sensor attachment on the human body either internally or externally. The goal is to assess the fatigue state of the human in a timely and accurate manner without any explicit communication from the human such that proactive measures can be taken to avoid an accident. This implicit communication not only enriches human's experience with a robot co-worker, it also assures a higher level of safety. We experiment with 18 subjects, where a force sensor mounted on the corner of a table-top mimics a robotic hand with a force sensor. Based on the experiments, we observed that force sensor value decreases as the subject gets fatigue. However, the change in force varies significantly for different subjects depending upon their physical strength. Moreover, classifying three states of fatigue instead of one makes it even more challenging. In spite of these challenges, we found that a variation of force change using a moving window based averaging can be utilized for detection of fatigue states with high accuracy, even in the presence of noise from perturbation of hand. Future research direction can include discriminating noise generated from the robot's mobility. Even though it is not a perfect system, the three state detection provides enough time to prepare for safety measures. Moreover, the deviation of detection time as compared to the ground truth is tolerable and in fact is gain compared to intrusive sensing, which is not suitable for seamless human-robot interaction.

References

[1] M. R. Al-Mulla, F. Sepulveda, and M. Colley. A review of non-invasive techniques to detect and predict localised muscle fatigue. *Sensors*, 11(4):3545–3594, 2011.

[2] M. R. Al-Mulla, F. Sepulveda, M. Colley, and A. Kattan. Classification of localized muscle fatigue with genetic programming on semg during isometric contraction. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 2633–2638. IEEE, 2009.

[3] J. Baraglia, M. Cakmak, Y. Nagai, R. Rao, and M. Asada. Initiative in robot assistance during collaborative task execution. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, pages 67–74. IEEE Press, 2016.

[4] F. Bernardo, R. Martins, and J. Guedes. Muscle fatigue assessment in manual handling of loads using motion analysis and accelerometers: A short review. In *Occupational Safety and Hygiene VI: Proceedings of the 6th International Symposium on Occupation Safety and Hygiene (SHO 2018), March 26-27, 2018, Guimarães, Portugal*, volume 31, page 67. CRC Press, 2018.

[5] R. Bhradwaj, M. Jagannath, and V. Balasubramanian. Changes in steering wheel grip interface pressure during simulated driving. *ICBSII-2013 (March 14 th-16 th)*, page 221, 2017.

[6] M. G. Boocock, G. A. Mawston, and S. Taylor. Age-related differences do affect postural kinematics and joint kinetics during repetitive lifting. *Clinical Biomechanics*, 30(2):136–143, 2015.

[7] N. Brown, S. Bichler, M. Fiedler, and W. Alt. Fatigue detection in strength training using three-dimensional accelerometry and principal component analysis. *Sports biomechanics*, 15(2):139–150, 2016.

[8] D. B. Chaffin. Localized muscle fatigue-definition and measurement. *Journal of Occupational and Environmental Medicine*, 15(4):346–354, 1973.

[9] V. Chandel, S. Mukhopadhyay, D. Jaiswal, D. S. Jani, S. Khandelwal, and A. Pal. C2p: An unobtrusive smartwatch-based platform for automatic background monitoring of fatigue. In *International Workshop on Human-centered Sensing, Networking, and Applications*. ACM, 2017.

[10] T. C. Chieh, M. M. Mustafa, A. Hussain, E. Zahedi, and B. Y. Majlis. Driver fatigue detection using steering grip force. In *Research and Development, 2003. SCORED 2003. Proceedings. Student Conference on*, pages 45–48. IEEE, 2003.

[11] H. Isa, S. R. Kamat, A. Rohana, A. Saptari, M. Shahrizan, et al. Analysis of muscle activity using surface electromyography for muscle performance in manual lifting task. In *Applied Mechanics and Materials*, volume 564, pages 644–649. Trans Tech Publ, 2014.

[12] Q. Ji, Z. Zhu, and P. Lan. Real-time nonintrusive monitoring and prediction of driver fatigue. *IEEE transactions on vehicular technology*, 53(4):1052–1068, 2004.

[13] K. LEE, S.-A. HYUN, and S. OAH. Detecting driver fatigue by steering wheel grip force. *International Journal of Contents*, 12(1):44–48, 2016.

[14] L. Peternel, N. Tsagarakis, D. Caldwell, and A. Ajoudani. Robot adaptation to human physical fatigue in human–robot co-manipulation. *Autonomous Robots*, Nov 2017.

[15] C. Ray, F. Mondada, and R. Siegwart. What do people expect from robots? In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3816–3821. IEEE, 2008.

[16] E. Spyropoulos, E. Chroni, P. Katsakiori, and G. Athanassiou. A quantitative approach to assess upper limb fatigue in the work field. *Occupational Ergonomics*, 11(1):45–57, 2013.

[17] A. Subasi and M. K. Kiymik. Muscle fatigue detection in emg using time–frequency methods, ica and neural networks. *Journal of medical systems*, 34(4):777–785, 2010.

[18] Q. Wang, J. Yang, M. Ren, and Y. Zheng. Driver fatigue detection: a survey. In *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, volume 2, pages 8587–8591. IEEE, 2006.