Linear Predictive Coding for Acute Stress Prediction from Computer Mouse Movements

Lawrence H. Kim[1a], Rahul Goel[2a], Jia Liang[3a], Mert Pilanci[4], and Pablo E. Paredes[1]

Abstract—**Prior work demonstrated the potential of using the Linear Predictive Coding (LPC) filter to approximate muscle stiffness and damping from computer mouse movements to predict acute stress levels of users. Theoretically, muscle stiffness and damping in the arm can be estimated using a mass-spring-damper (MSD) biomechanical model. However, the damping frequency (i.e., stiffness) and damping ratio values derived using LPC were not yet compared with those from a theoretical MSD model. This work demonstrates that the damping frequency and damping ratio from LPC are significantly correlated with those from an MSD model, thus confirming the validity of using LPC to infer muscle stiffness and damping. We also compare the stress level binary classification performance using the values from LPC and MSD with each other and with neural network-based baselines. We found comparable performance across all conditions demonstrating LPC and MSD model-based stress prediction efficacy, especially for longer mouse trajectories.**

Clinical relevance— **This work demonstrates the validity of the LPC filter to approximate muscle stiffness and damping and predict acute stress from computer mouse movements.**

I. INTRODUCTION

Stress is an instrumental factor for the emotional, cognitive, and physical well-being of people. A large corpus of research has demonstrated strong links between stress and a wide range of chronic health risks such as cardiovascular disease [1], diabetes [2], hypertension [2], obesity [2], and coronary artery disease [3]. Physiological reactions induced by stress are symptomatic of mental illnesses such as anxiety disorder and depression, which are a leading cause of suicides [4]. Chronic stress can also lead to mood swings, social isolation, and even reduction in academic achievements among adolescents [5].

Re-purposing data from everyday use human-machine interfaces have been proposed to continuously monitor and detect acute stress levels of individuals [6], [7]. While there are a plethora of data streaming into our smart devices that could be useful for stress prediction, movement data is of particular interest due to its privacy-preserving nature compared to other types of data such as app usage, messages,

[a]These authors contributed equally

[1]Lawrence H. Kim and Pablo E. Paredes are with the Department of Psychiatry and Behavioral Sciences, Stanford University, Stanford, CA, USA {lawkim,pparedes}@stanford.edu

[2]Rahul Goel was with the Department of Radiology, Stanford University, Stanford, CA, USA rahulg@alum.mit.edu

[3]Jia Liang is with the Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA, USA jialiang@stanford.edu

[4]Mert Pilanci is with the Department of Electrical Engineering, Stanford University, Stanford, CA, USA pilanci@stanford.edu
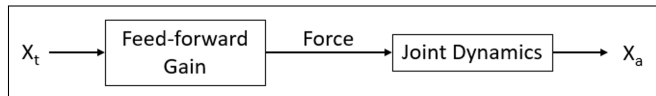
Fig. 1. Open-loop feedforward model for rapid goal-directed arm movement.

and location. Sun et al. demonstrated in the MouStress paper, the effectiveness of using computer mouse motion data to measure acute stress where the damping frequency (square of which is proportional to stiffness) values obtained from a linear predictive coding (LPC) filter, were larger under a stressed condition than a calm condition [6]. The steering wheel of a car simulator has also been successfully shown to detect acute stress, with data from only a few turns [7]. However, the parameters obtained from the LPC filter, in prior studies [6],[7], were not yet compared with corresponding parameters obtained from a theoretical mass-spring-damper (MSD) biomechanical model of the arms. This study aims to bridge that gap to help progress toward continuous stress monitoring using everyday mouse movement data.

II. BIOMECHANICAL MODEL OF THE ARM

A. Mass-Spring-Damper (MSD) Model

In the context of computer mouse interactions, what precedes clicking is a rapid, goal-directed skilled movement. This skilled movement is primarily open-loop in nature and is influenced by feedforward use of sensory information obtained in previous computer mouse movements and experience [9]–[13]. The dynamic behavior during any routine skilled movement is influenced by passive joint properties, initial motor commands, sensory feedback, mechanical constraints, and mechanics of the neuromuscular system, aka, biomechanics [14]. Research in biomechanics has shown that a rapid goal-directed movement can be modeled as a step response of a linear second-order system, i.e., a classic mass-spring-damper system [9]–[13]. Typically, smoothed raw movement data is fit to a simple open-loop feedforward model [12], [13].

Fig. 1 shows the model describing the relationship between the target (X_t) and the actual (X_a) computer mouse position in the horizontal direction. The open-loop transfer function between X_t and X_a is given by Eq. 1 as:

$$\frac{X_a(s)}{X_t(s)} = \frac{K_f}{Js^2 + Bs + K} \quad (1)$$

where, K_f represents the feedforward gain, J represents the moment of inertia of the arm, B is the viscous damping

coefficient of the damper, and K is the stiffness of the spring. Eq. 1 can also be represented as:

$$\frac{X_a(s)}{X_t(s)} = \frac{K_p \omega^2}{s^2 + 2\omega\zeta s + \omega^2} \qquad (2)$$

where, K_p is the static gain, ω is the damping frequency, and ζ is the damping ratio. Parameters (K_p, ω, ζ) were estimated using an iterative time-domain identification technique called prediction-error minimization ("pem" from the System Identification Toolbox of MATLAB 2020b [Mathworks, Natick, MA]) that minimizes the cost function defined as the sum of squares of the difference between the actual and the simulated output. Normalized root mean square error was used to estimate the goodness of fit (GOF) of the model and is expressed as:

$$100 \times \left(1 - \frac{||X_a - X_s||}{||X_a - mean(X_a)||} \right) \qquad (3)$$

where, X_s is the simulated step response from the model.

B. Linear Predictive Coding (LPC) Filtering Technique

Linear predictive coding is a finite impulse response (FIR) filtering technique (Eq. 4) that builds a predictive model of future samples based only on linear combinations of observed signals from the past [15]. LPC model assumes an all-pole filter that can approximate systems where poles are dominating signal characteristics. It turns out that an ideal second-order system, such as the MSD system described earlier, which is an all-pole system in the Laplace domain, is similar in structure to that of a simple LPC model. That is, if we build an LPC model that best fits a series of samples, we can recover an approximation to the MSD parameters.

$$H(z) = \frac{1}{A(z)}, \ A(z) = \frac{E(z)}{X(z)} = 1 - \sum_{k=1}^{p} a_k z^{-k} \qquad (4)$$

where H(z) is the system response, a_k are the LPC coefficients, E(z) is the system approximation error, p is the order of the approximation.

In our previous two studies [6] and [7], we used an LPC filter of order 4 that generates a sequence of 4 coefficients. Then, we constructed a 4th order polynomial from the 4 coefficients and considered the complex roots of it (that will exist in the case of an under-damped system). Then we estimated, the damping frequency (ω) as the imaginary part of the complex root ($\omega = |I(r)|$) , and the damping ratio (ζ) as the ratio of the complex root's real part and its absolute value ($\zeta = \frac{|R(r)|}{||r||}$). We claimed that the equation of complex roots is similar to the characteristic polynomial of the 2nd order MSD system. Thus ω and ζ obtained from each of the two approaches (LPC and MSD) should be correlated. The primary aim of this study is to provide evidence towards that claim by showing the correlation between the parameters from the two approaches. It is important to note that the LPC approach is substantially simpler and lends itself to efficient real-time hardware and software implementations.
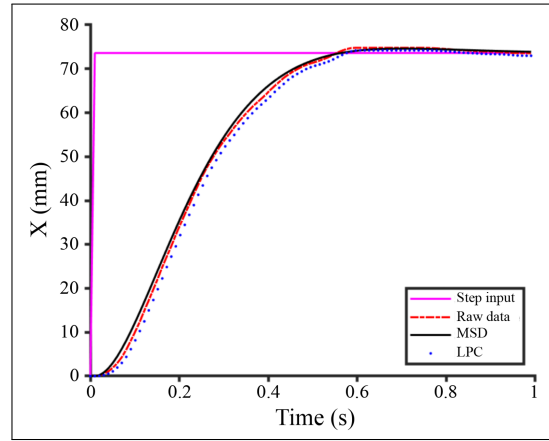

Fig. 2.   An example of raw data compared with the responses from MSD and LPC.

III. METHODOLOGY

Using the dataset from the MouStress study [6], we computed the damping frequency, ω, and damping ratio, ζ, using the MSD model and LPC filter. Fig. 2 shows an example of the resulting responses from MSD and LPC compared with the raw signal. Then, we computed the correlation between the ω and ζ from both methods after removing outliers and compared the classification performance when using LPC and MSD-based ω and ζ, either as individual parameters or in combination.

A. Dataset

We used the data from the point-and-click task during the MouStress study [6], as clicking is the most frequent computer mouse event [8]. As the MSD model assumes rapid goal-directed movement, we primarily leverage the mouse movement data before the click to compute the ω and ζ using MSD and LPC. As described in the MouStress paper [6] and shown in Fig. 3, the objective of the user was to move horizontally and click the two targets in succession (first, green on the left, and then, blue on the right) as quickly and accurately as possible. Since the primary movement direction was horizontal. we only looked at the x-axis movement data. Two task parameters, distance D (64px, 128px, 256px, 512px, 1024px) and width W (8px, 16px, 32px, 64px), were varied. Each of the N = 49 participants (26 female and 23 male, mean age = 20) performed 5 repetitions of the task with the same configuration under both stressed and calm conditions. The stressed conditions were elicited through 5 minutes of recursive mental math calculations. More details of the experimental design can be found in the MouStress paper [6].

B. Correlation Analysis

To find correlations between the ω's and ζ's obtained from each of the MSD and LPC approaches, we conducted Spearman's rank correlations for different thresholds of GOF ranging from 0% to 95%.
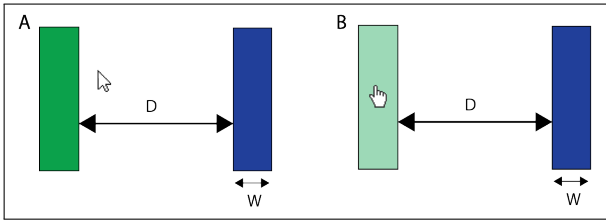
Fig. 3.   Point-and-click task from MouStress [6]. a) Participants have to reach the green region and b) click it, which is then dimmed to provide feedback.

Fig. 4.   Architecture of the methods based on A) LSTM and B) CNN. (FC = fully connected layer, ReLU = rectified linear unit)

C. Binary Stress Classification

We compared performance using eight different machine learning (ML) models. Six models were based on a standard ML technique, called the support vector machines (SVMs), and included models built with either individual parameters (ω or ζ), or combination of parameters (ω and ζ), derived from each of the MSD and LPC approaches. We also deployed two neural network-based ML methods, specifically Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN), using smoothed raw computer mouse movement data. Our goal was to compare the performance (in terms of accuracy) from MSD and LPC-based methods with neural network baselines, which do not leverage any domain knowledge or processing. To minimize the effects of inter-subject differences and the distance of point-and-click tasks on the performance of the tasks, we decided to build a different classifier per participant per distance of the point-and-click task. This led to each classifier having 40 data samples (20 trials for stressed condition and 20 trials for calm). Based on the data, we did not see significant influence of the width and thus did not take that into consideration. SVM was chosen as some of our initial analyses comparing different standard ML classifiers like SVM, Decision Tree etc., with part of the data, consistently showed that SVM outperformed other standard ML classifiers.

For each trial, the dataset consisted of the x-position of the computer mouse obtained at a fixed frequency (2 kHz). As a result, input to the LSTM and CNN classifiers were time-series, and output was the binary stress condition (either stressed or calm). Since the same participant in each trial might have finished the point-and-click task at different times, the length of the time-series data was variable. Thus, we first tried to use a baseline LSTM network-based method. However, the direct application of the LSTM model on the raw data turned out not to be successful (based on non-decreasing validation loss). Further investigation of the raw data revealed that for all trials, the final value of x position of each time series was generally repeated many times (which was obvious as the location of the target bar was fixed). As a result, the last several points in each time-series looked very similar regardless of the condition. Thus, we let the LSTM network learn from windowed time-series data with the end points removed. To implement this, we fixed a cutoff value (representing the last points of the new time-series) for all the point-and-click tasks for the same distances. The cutoff
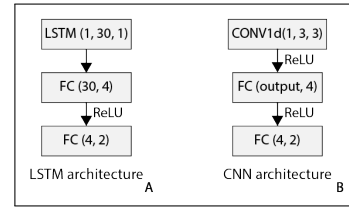
values were set to be 100 ms for 64x, 125 ms for 128x, 150 ms for 256x, 350 ms for 512x, and 500 ms for 1024x. To make a fair comparison, the same cutoff values applied to inputs of the LSTM model were also applied to that of the CNN model, which is described below.

Due to the scarcity of our data, the final version of the LSTM model we used consisted of only one LSTM layer followed by two fully-connected layers, and its architecture is shown in Fig. 4A. The LSTM-based model was trained with Stochastic Gradient Descent Optimizer with cross-entropy loss as the loss function. The batch size was equal to 4, and the learning rate was fixed at 0.05. We split our data such that 80% were used for training and validation, and the remaining 20% for evaluation. The model was trained from scratch for 100 epochs, and the model with the best validation accuracy during training was saved and used to infer the test accuracy. This process was repeated 10 times for each subject and each distance.

The final architecture of the CNN model is shown in Fig. 4B. The CNN model was trained exactly like the LSTM based model as described above with a learning rate of 0.001, but was trained for 500 epochs. Both the neural-network models were implemented in Pytorch 1.7.0.

For the standard ML models, the ω and ζ parameters obtained from each of MSD and LPC approaches were fit with the SVM classifier (either individually or in combination) with default hyperparameters and min-max normalization. SVM was imported from the sklearn library. The accuracy was obtained from averaging accuracies from 5-fold cross-validation for each iteration. The overall process (extraction of parameters with either MSD or LPC, fitting SVM, obtaining classification accuracy) described above was repeated 10 times, and the average results across the 10 iterations are reported. We also compare the classification performance from the LPC and MSD-based methods using paired t-tests after confirming normality via Kolmogorov-Smirnov test of normality.

IV. RESULTS & DISCUSSION

The means and standard errors of ω and ζ from MSD and LPC under calm and stressed conditions are computed after removing outliers, where $\zeta \leq 0$ or $\zeta > 100$, and are shown in Table I. We see a significant ($p < 0.001$) increase in ω of the stressed condition relative to the calm condition. However, although the ζ from the LPC was significantly higher under stressed condition, than calm (as previously shown in [6]),

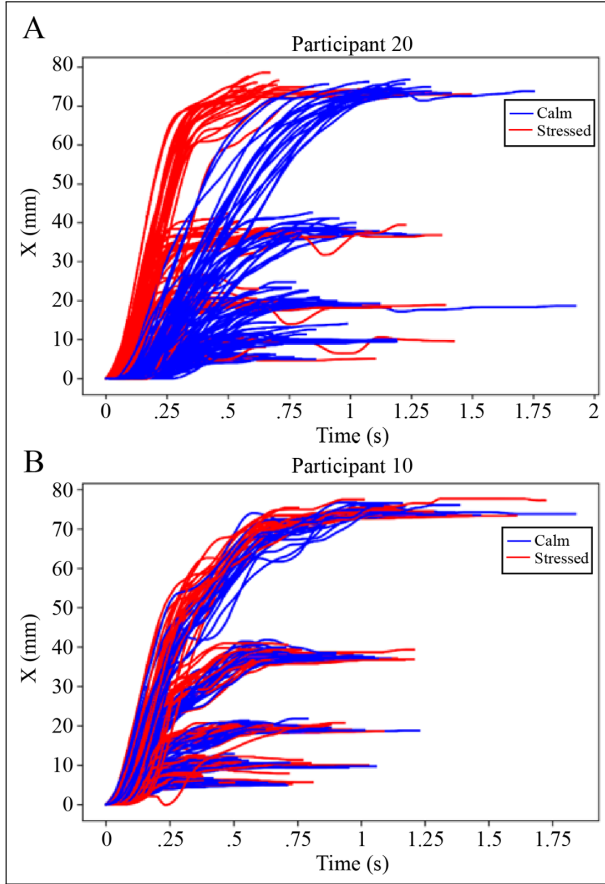|	ω			
Model	calm	stressed	t(48)	p
MSD	12.9 (0.5)	14.4 (0.4)	3.6	< 0.001*
LPC	0.261 (0.002)	0.268 (0.001)	3.8	< 0.001*
	ζ			
Model	calm	stressed	t(48)	p
MSD	1.00 (0.03)	.97 (0.03)	1.1	0.28
LPC	0.5635 (0.0005)	0.5652 (0.0005)	3.2	0.002*

Fig. 5. Examples of mouse pointer trajectories from two participants. a) One with, and b) one without, visually clear boundaries between the stressed (Red) and calm (Blue) conditions.

the ζ from MSD tended to be lower in stressed condition than calm (as previously hypothesized in [6]).

Fig. 5 show examples of all the raw mouse motion trajectories from two representative participants under stressed and calm conditions. Fig. 5A demonstrates computer mouse trajectories with clear visual distinction between the stressed and calm conditions. Such trends were noticed visually for 16 of the 49 participants when the distance between the bars was 1024x. Fig. 5B shows a participant's data where no visually clear boundary could be drawn between the trajectories under stressed and calm conditions. 33 participants exhibited similar behavior.
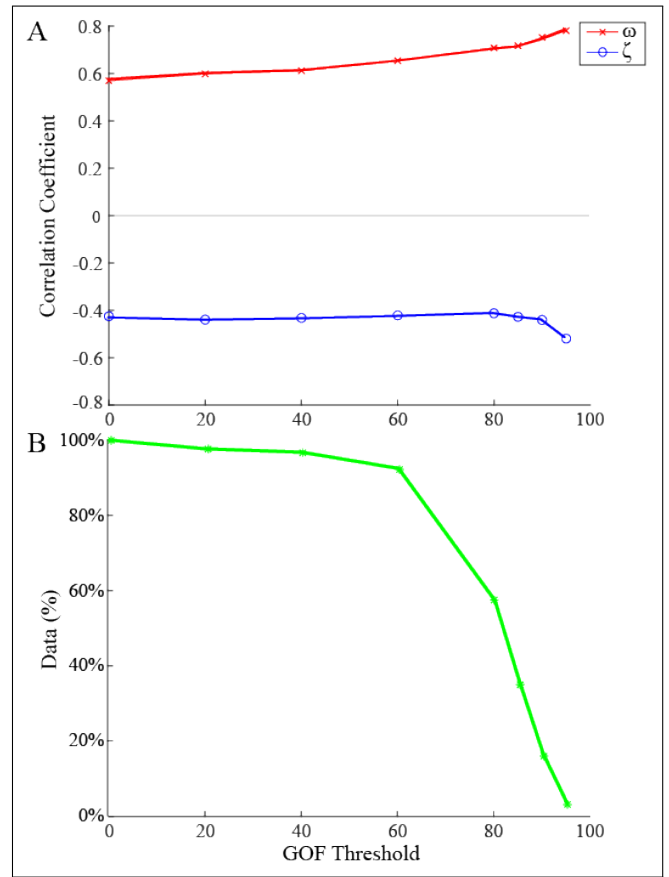

Fig. 6. A) The correlation coefficients between the ω and ζ from MSD and LPC for different thresholds of GOF are shown with red and blue lines, respectively. B) The green line indicates the percentage of the corresponding data according to different thresholds of GOF.

A. Correlation

Fig. 6A show the correlations between the damping frequencies and damping ratios from MSD and LPC for different thresholds of GOF. All correlations were statistically significant ($p < .001$). The correlation coefficient between the damping frequencies was strong and reached 0.7 with GOF of 80% and went even higher for data with higher GOF. These results suggest that the damping frequency, as obtained from the LPC filter could be a very good proxy for the damping frequency, as obtained from a theoretical MSD model. For the damping ratios, we saw moderate correlations of approximately -0.45, even for data with GOF of 90%. The negative correlation between the damping ratios suggests, that the damping ratio from LPC (although significantly different for stressed and calm conditions in our task), is not that good a proxy for damping ratio obtained in a traditional sense from a theoretical MSD model. Further, as the threshold for GOF reached 80%, we saw a steeper decline in the percentage of data that met the requirement as shown in Fig. 6B, where only around 55% of the entire data met the 80% GOF threshold criteria, typically common in system modeling studies.

Distance	CNN	LSTM	MSD ω,ζ	LPC ω,ζ	MSD ω	LPC ω	MSD ζ	LPC ζ
64x	51.8% (0.65%)	54.7% (1.1%)	58.8% (1.9%)	53.5% (1.4%)	**60.0%** (1.9%)	54.6% (1.7%)	51.2% (1.0%)	57.1% (1.7%)
128x	53.6% (0.95%)	56.4% (1.5%)	56.4% (2.0%)	55.9% (1.8%)	57.1% (2.0%)	58.8% (1.0%)	52.3% (2.0%)	**58.9%** (2.0%)
256x	55.1% (1.04%)	58.3% (1.7%)	59.1% (2.0%)	57.2% (1.6%)	59.5% (2.0%)	**62.3%** (1.8%)	55.0% (1.3%)	60.4% (1.9%)
512x	60.1% (1.5%)	65.3% (1.8%)	60.0% (2.2%)	58.6% (1.3%)	60.7% (2.1%)	**67.2%** (2.0%)	54.9% (1.6%)	64.4% (2.1%)
1024x	62.2% (1.6%)	67.3% (2.0%)	63.6% (2.3%)	61.4% (1.4%)	65.3% (2.2%)	**72.9%** (1.8%)	56.0% (1.6%)	69.4% (1.9%)
Overall	56.6% (1.01%)	60.4% (1.4%)	59.6% (1.8%)	57.3% (1.3%)	60.5% (1.7%)	**63.2%** (1.4%)	53.9% (0.7%)	62.0% (1.6%)

B. Binary Acute Stress Classification

Table II shows the classification accuracies from the different ML models. The LPC ω-based model had the highest overall accuracy, followed by LPC ζ-based model, MSD ω-based model, and LSTM-based model, respectively. Paired t-tests showed that LPC-based methods had significantly higher classification accuracies than MSD-based methods when either using ω ($p = .01$) or ζ ($p < .001$). This result suggests that the LPC-based models can match or outperform MSD-based models and neural network-based baseline models for classifying acute binary stress levels. Specifically, the classifier based on the LPC-based ω yielded accuracy above 75% for nearly half of the participants (23 of 49) for trials with 1024x distance. The relevance of LPC-based ω was also demonstrated in prior literature [6],[7] . In the future, it may be worth exploring whether using only the data samples above certain GOF (for MSD model) or below certain error variance (for LPC filter, and something not considered in this study) improves the accuracies of the MSD- and LPC- based classifiers, respectively. Further, it is possible that hyperparameter tuning in SVM would yield better classification accuracies. For the neural network-based classifiers, training with larger amount of data may yield higher classification accuracies.

It is also worth noting the reduction in accuracy observed with smaller distances between targets. A possible explanation for this reduction is that as the targets are closer together, there is less interaction from the larger muscles in the arm versus the hand. As we progress towards in-the-wild studies, it would be relevant to investigate optimal window lengths further to apply MSD or LPC-based estimators.

V. CONCLUSION

In this paper, we demonstrate that the parameters (especially damping frequency, aka stiffness) derived from an LPC filter are valid and a good proxy (aka strongly correlated) to those derived from a biomechanical MSD model of the human arm to predict binary acute stress levels of users based on their computer mouse movement data. We also compared stress classification performance using SVM to that of neural network-based models such as LSTM and CNN, and found that MSD and LPC-based models produced higher classification accuracies than LSTM and CNN-based models. This demonstrates the potential of LPC and MSD models in predicting the acute stress levels of users from their computer mouse movement data to enable continuous stress monitoring. In the future, we plan to explore using a combination of neural network-based approaches with MSD or LPC, applying those methods to analyze in-the-wild data, and combining physiological signals to further improve the classification performance.

REFERENCES

[1] P. L. Schnall, P. a Landsbergis, and D. Baker, "Job Strain and CARDIOV ASCULAR DISEASE Peter," Annu. Rev. Public Health, no. 9, pp. 381–411, 1994.

[2] D. N. Brindley and Y. Rolland, "Possible Connections between Stress, Diabetes, Obesity, Hypertension and Altered Lipoprotein Metabolism that may Result in Atherosclerosis," Clinical Science, vol. 77, no. 5, pp. 453–461, Nov. 1989.

[3] T. G. Pickering, "Mental stress as a causal factor in the development of hypertension and cardiovascular disease," Current Hypertension Reports, vol. 3, no. 3, pp. 249–254, 2001.

[4] G. M. Slavich and R. P. Auerbach, "Stress and its sequelae: Depression, suicide, inflammation, and physical illness.," APA handbook of psychopathology: Psychopathology: Understanding, assessing, and treating adult mental disorders (Vol. 1)., vol. 1, pp. 375–402, 2018.

[5] K. Schraml, A. Perski, G. Grossi, and I. Makower, "Chronic Stress and Its Consequences on Subsequent Academic Achievement among Adolescents," Journal of Educational and Developmental Psychology, vol. 2, no. 1, pp. 69–79, 2012.

[6] D. Sun, P. Paredes, and J. Canny, "MouStress: Detecting stress from mouse motion," Conference on Human Factors in Computing Systems - Proceedings, pp. 61–70, 2014.

[7] P. E. Paredes, F. Ordoñez, W. Ju, and J. A. Landay, "Fast furious: Detecting Stress with a car steering wheel," Conference on Human Factors in Computing Systems - Proceedings, vol. 2018-April, pp. 1–12, 2018.

[8] M. Antal and E. Egyed-Zsigmond, "Intrusion detection using mouse dynamics," IET Biometrics, vol. 8, no. 5, pp. 285–294, 2019.

[9] J. Müller, A. Oulasvirta, and R. Murray-Smith, "Control theoretic models of pointing," ACM Transactions on Computer-Human Interaction, vol. 24, no. 4, 2017.

[10] F. Fischer, A. Fleig, M. Klar, L. Gruene, and J. Mueller, "An Optimal Control Model of Mouse Pointing Using the LQR," 2020, [Online]. Available: http://arxiv.org/abs/2002.11596.

[11] S. Aranovskiy, R. Ushirobira, D. Efimov, and G. Casiez, "A switched dynamic model for pointing tasks with a computer mouse," Asian Journal of Control, vol. 22, no. 4, pp. 1387–1400, 2020..

[12] R. Goel and W. H. Paloski, "Motor Control Performance During Rapid Voluntary Movements of Elbow and Knee," Journal of Motor Behavior, vol. 48, no. 4, pp. 348–356, 2016.

[13] L. B. Bagesteiro and R. L. Sainburg, "Nondominant arm advantages in load compensation during rapid elbow joint movements," Journal of Neurophysiology, vol. 90, no. 3, pp. 1503–1513, 2003.

[14] F. Popescu, J. M. Hidler, and W. Z. Rymer, "Elbow impedance during goal-directed movements," Experimental Brain Research, vol. 152, no. 1, pp. 17–28, 2003.

[15] D. O. Shaughnessy, "Linear predictive coding," IEEE Potentials, vol. 7, no. 1, pp. 29–32, 1988.

[16] T. Söderström and and P. Stoica, "System Identification," Prentice-Hall International, 1989.