Feedback Control of Human Stress with Music Modulation

Andrew Fu

April 18, 2020

Advisor: Prof. Kevin Passino

A thesis presented for the degree of
Undergraduate in Electrical & Computer Engineering

Department of Electrical & Computer Engineering
The Ohio State University
Columbus, Ohio

1 Abstract

Mental stress has known detrimental effects on human health, however few algorithmic methods of reducing mental stress have been widely explored. While the act of listening to music has been shown to have beneficial effects for stress reduction, and furthermore, audio players have been designed to selectively choose music and other inputs with the intent of stress reduction, limited work has been conducted for real-time stress reduction with feedback control using physiological input signals such as heart rate or Heart Rate Variability (HRV). This thesis proposes a feedback controller that uses HRV signals from wearable sensors to perform real-time (< 1 second) modulations to music through tempo changes with the goal to regulate and reduce stress levels. A standardized, stress inducing test based on the popular Stroop test is also introduced, which has been shown to induce acute stress in subjects and can be used as a testing benchmark for controller design. Ultimately, a controller is presented that when used is not only able to maintain stress levels during stress-inducing inputs to a human but even provides de-stressing effects beyond baseline performance.

2 Introduction

2.1 Literature Review

2.1.1 Stress and HRV

Mental stress has widely been studied in various fields because of its significant impact on daily human life. Stress has been shown to lead to harmful effects such as decreased attention spans, impaired cognitive ability and memory functions, and negatively influenced decision making and judgement capabilities [1, 2, 3, 4, 21]. Because of the detrimental effects on health and mental cognition that stress brings, a large focus in research has been identifying physiological signals and measurement techniques that can provide an accurate representation of stress within humans. In the literature, signals including galvanic skin response, blood pressure, respiration rate, heart rate (HR) through electrocardiograms (ECG) or photoplethysmographs (PPG), and heart rate variability (HRV) have been explored for their potential in representing stress [21]. Among these signals, HRV, which describes the variability in heart rate by measuring the successive changes in inter-beat intervals, the time measured from consecutive heartbeats by finding the peak-to-peak differences in an ECG or PPG signal, has been commonly accepted as a metric for acute stress levels in an individual [5]. Moreover, multiple HRV metrics have been experimentally shown to change significantly during stressful events, most notably exhibiting more profound changes than metrics such as heart rate [20, 23].

HRV provides a baseline intuition when measuring stress: a healthy heart beat is not steady – large beat-to-beat fluctuations and a relatively high amount of heart rate variation are associated with lower levels of long- and short-term stress [6]. HRV has also been found to be strongly correlated with the human aging process, generally decreasing with age [7]. HRV metrics have also been linked to the parasympathetic nervous system (PNS) and the sympathetic nervous system (SNS), the former being responsible for physiological regulation while the latter being responsible for acute stress levels [8]. HRV metrics are divided into two subcategories: time- and frequency-domain metrics [9]. Time domain metrics rely on the calculation of "RR-peaks", or the distance in time between successive peaks in an ECG or PPG signal, which correspond to the time differences between successive heartbeats. These peaks are then filtered for outliers and normalized into "NN-peaks", which result in generation of metrics such as MeanNN, SDNN, and RMSSD, representing average NN-intervals, standard deviation of NN-intervals, and root mean square difference of NN-intervals, respectively. Frequency domain metrics transform the time-domain signal from an ECG or PPG to a frequency domain signal, separating the signal into four frequency bands: ULF (ultra low frequency) $<$0.0033 Hz, VLF (very low frequency) 0.0033-0.04 Hz, LF (low frequency) 0.04-0.15 Hz, and HF (high frequency) 0.15-0.4 Hz [21].

The majority of HRV analysis and its link to acute stress levels has been conducted for long term (24-hour) processes, and even the most extreme short term analysis still relies generally on five-minute processes [19]. For long-term and short-term HRV analysis, HRV metrics in both the time- and frequency-domain have been shown to change significantly during stressful events [22, 20], however significantly less research has been conducted on the response of ultra-short term ($<$five minute) HRV metrics to stress. For real-time stress measurement and regulation, however, ultra-short metrics are necessary in order to prevent extensive and potentially detrimental delays in a control loop; ultimately, the choices of an ultra-short term window length and specific HRV metrics are extremely important for stress measurement validity.

2.1.2 Stress and Music

Music represents a naturally intuitive choice for feedback control of stress because it is non-intrusive, can be dynamically modified in real time, and has received significant attention in research for its potential benefits in reducing acute stress levels. Simply listening to music has not only been found to impact the psychobiological stress system, but it has been linked to a faster recovery time after a stressful event, reduced levels of the endocrine response (lower cortisol levels), and less stimulation of the sympathetic nervous system [10, 11]. In addition, it has been shown experimentally that specific features of music including valence and arousal, metrics that describe the pleasure and excitement of a particular song, respectively, each have specific effects on stress reduction and can provide a framework for not only reducing stress levels in individuals but tailoring a more effective, individualized music therapy approach [12].

Generally, low-arousal and high-pleasure music has been found to be more effective in reducing anxiety and stress levels when compared to high-arousal and low-pleasure music [13, 14, 15, 16]. At a lower level, higher pitches, faster tempos, and louder volumes are associated with stressful events, while lower pitches, slower tempos, and softer volumes are associating with calming events [17]. Using these high-level principles, and combined with physiological signals including HRV, previous "affective music players" have been developed and explored that use SDNN as an input to determine the type of music to be played in an attempt to use rule-based feedback control to regulate HRV and stress [18]. The work regarding an "affective music player", however, focuses on the choice of an entire song. This thesis, however, focuses on modulating via feedback control a song while it is playing. Figure 1 provides an illustration of this proposed control loop.

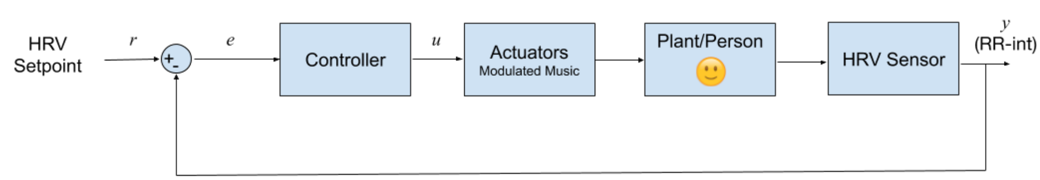

Figure 1: Proposed controller loop.

2.2 Overview of Thesis

This document overviews research into the field of feedback control of stress through music. The main motivating factor for much of the work conducted for this thesis is to lay the foundation for more detailed analysis into controller development and design for this application, and focuses on two main problems: designing a methodology to obtain a metric of stress from heart rate data, and designing an experiment to evaluate controllers and their success with music modulation. To accommodate the former goal, a large portion of this document is concerned with establishing a link between HRV metrics, such as SDNN, and the perceived stress level of an individual. For the latter goal, this document provides detailed instructions on setting up an experiment to evaluate controller design by providing a toolbox of tests, metrics, and routines. This includes algorithms for real time, online music modulation in MATLAB, data acquisition wrappers for wearable sensors such as the Empatica E4, an overview of techniques to calculate HRV metrics in MATLAB, and an automated web application for administering a standardized stress-inducing test based on the Stroop Test. It also defines a standard for controller success and performance in reducing perceived stress levels.

Using this toolbox, this thesis also overviews some initial results with designing a feedback controller that is able to regulate HRV and stress levels, and reduce the overall impact of stress via music modulation. This includes a system identification motivated analysis on the effects of various features of music, such as tempo and volume, on HRV metrics, as well as a brief treatment of controller development, design, and tuning with proportional integral derivative (PID) control. Finally, this thesis overviews several directions for potential future work.

3 Methodology

Feedback control with music modulation creates a unique problem with HRV as a control input because of how HRV is measured. Even short-term HRV metrics at five minute intervals generate potential problems: longer window lengths generate delays in the controller response. Considering that many songs are often only two to three minutes in duration, a five minute window size may not even lead to any modulation of the song.

This problem has lead to the exploration of ultra-short term HRV analysis, which consists of using window lengths less than five minutes, and in some cases, as short as 30 seconds. These ultra-short term HRV metrics provide a more reactive, but noisy HRV response from the subject, yet facilitate real time feedback control. This document overviews the choice of the ultra-short term HRV window length of 120s, as well as the choice of SDNN as the HRV metric under control.

3.1 HRV Window Length

Ultra-short term HRV metrics are defined by Castaldo et al. as HRV metrics, such as SDNN or RMSSD, measured with less than five minutes of PPG data [20].

For feedback control, an ultra-short term HRV metric determines the window length in which PPG data is used for feedback and analysis. For the purposes of feedback control experiments, window lengths including 10s, 20s, 30s, 60s, 120s, 180s and 240s were explored. An important consideration for window length in the ultra-short term range is how well a particular window length serves as a surrogate for a five minute HRV metric, which has previously been shown to be heavily correlated to, and act as a good predictor for, actual stress levels [21, 22]. As such, a five minute HRV metric was considered the "gold standard" to compare metrics with different window lengths to.

An important consideration that Castaldo et al. points out with regards to HRV feature calculation is that certain metrics, specifically LF, HF, LF/HF and total power, among other frequency domain features, cannot be computed for certain extremely short term window lengths. Spectral analysis is recommended to be performed with a window length that is at least 10 times longer than the lowest frequency component.

The low frequency (LF) band for HRV ranges between 0.04-0.15Hz, which translates to a period length of 6.7s to 25s, implying that for the LF band to be properly captured, a minimum sampling window length of between 67 - 250s is required. Generally, the literature recommends a window length of no less than 120s for LF (and therefore LF/HF ratio and total power), and a window length of no less than 60s for HF [19], however, to obtain the total power spectrum of the HRV signal which also includes an ultra-low frequency component (ULF) from 0.003-0.04Hz, a window length of upwards of 3300s (55 min) can be required.

Castaldo et al.'s main findings are that six out of the 23 HRV features under analysis showed consistency as surrogates for short-term HRV in all window lengths tested, from 1 min to 5 min with 1 min increments. These six HRV features - MeanNN, StdNN, MeanHR, StdHR (standard deviation of HR), HF, and SD2 (Poincaré plot ellipse length), are corroborated by results from Munoz et al. which tested ultra short term RMSSD and SDNN as surrogates for short term HRV [23]. Munoz et al., like Castaldo et al., used the five minute, short-term HRV window length as the "gold standard" to which ultra-short term metrics were compared, finding that a 120s window length for SDNN and RMSSD had a near perfect correlation with the gold standard, with correlation coefficients $r = 0.956$ for SDNN and $r = 0.986$ for RMSSD. This information, combined with the results from Castaldo et al. and the analysis of frequency domain HRV metrics, suggests strongly a window length of 120s for feedback control, which also mitigates the problem of long delays in the controller response as well.

3.2 HRV Metric Choice

From the decision to use a window length of 120s, the next natural question for feedback control with HRV is which metric to use as a controller input. Two main concerns arise: first, which metric serves as the best indicator for actual stress levels, and second, which metric can be used in the ultra-short term as a suitable surrogate for short term HRV analysis. The six metrics that Castaldo et al. cited as suitable for ultra-short term HRV analysis – MeanNN, StdNN, MeanHR, StdHR, HF, and SD2 – can then each individually be evaluated as indicators for actual stress levels. StdNN (SDNN) was found by Castaldo et al. to decrease significantly ($p = 0.05$) during periods of stress in all window lengths from 30s - 5 min. This result is also evident in Delaney and Brodie, which found that SDNN dropped 7.5 ms from a 64 ms baseline, significant at the $p = 0.05$ level as well [22].

4 Experiment Overview

The results obtained by Delaney and Brodie suggest a framework for quantifying the success of a feedback controller in the loop. Their analysis elucidated the effects of a short term stress input on HRV metrics, including SDNN and LF/HF. The following five minute, modified "Stroop Word Color Conflict Test", with an additional mental mathematics component, was administered to half of the participants, while the other half served as a control group [24]:

1. Naming the text color of nonsensical words such as "ZYP" or "WORP"
2. Simple mathematical problems, such as 1+2-3+4-5=?
3. Traditional Stroop test [24]
4. Modified Stroop test, where the words shown were words with a traditional color association printed with a different text color (such as "GRASS" printed in red)

Subjects were told to answer questions as quickly and accurately as possible, and they were told that the subject with the highest number of correct answers would be given a cash prize [22]. Delaney and Brodie showed that SDNN dropped and average of 7.5 ms from a 64 ms baseline for the 15 participants in the induced stress group, supported further by showing that SDNN dropped only an average of 0.3 ms from a 60.5 ms baseline for the 15 participants in the control group. Therefore, a sufficient condition to ensure that a controller is having a positive effect on stress would be to show that when a controller and music are introduced, the SDNN for a group having to take a Stroop test does not drop significantly and moreover would be similar to the control group from Delaney and Brodie's analysis.

This result could be shown via the following 17 minute experiment:

1. Two minute buffer period to ensure that window length of 120s is satisfied (longer or shorter depending on if larger or smaller window lengths are used)
2. Five minute pre-test period
3. Five minute Stroop test period (following the procedure outlined by Delaney and Brodie)
4. Five minute post-test period

A five minute period length is chosen to ensure the validity of a short-term HRV metric. This experiment would then be repeated three times total: once for no music, once for music without a controller, and once for music modulated by a feedback controller. SDNN could then be sampled for the pre-, during-, and post-test periods to obtain information on whether or not the controller was actively able to maintain SDNN levels through the stress input. Because it is difficult, if not impossible, to entirely prevent subjects from experiencing immeasurable environmental stress inputs during the duration of the experiment, a true "no stress" period of comparison cannot realistically be obtained. On the other hand, it is possible to guarantee a period of stress using the stress test: therefore, the most important point of comparison between these three experiments will be during the five minute Stroop test period.

4.1 Stress Inducing Test Web Application

To facilitate the administration of a modified stress test that would automatically start and stop at predefined experimental times as well as record and relay performance information back to the controller, a fully modular, Angular 7.2 frontend web application, with a simple Node 10.16.3 backend microservice, was developed, as shown in Figure 2.

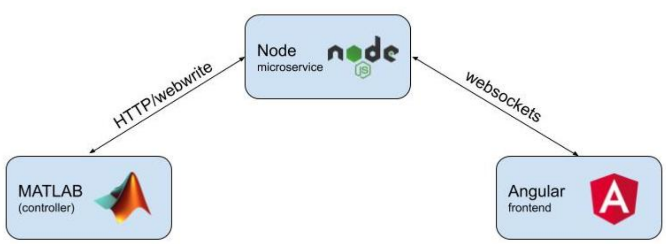

Figure 2: Web application architecture.

Information is passed between the controller, written in MATLAB, to the Stroop test Angular web application via the node microservice, which has a simple RESTful API to communicate with MATLAB and websockets to communicate with an Angular service on the frontend.

4.1.1 Node Microservice Architecture

The node microservice, on top of also serving the Angular frontend, implements the following GET API endpoints:

Table 1: API Endpoints

Endpoint	Parameters	Funtion
`/stroop`	Complexity, MaxResponseTime	Sets the test complexity and maximum response time
`/time`	tts (time to start)	used for a countdown clock to test start. When tts=0, the stress test begins
`/duration`	Test Duration	Sets test duration for each individual subtest

The node microservice was hosted on `localhost:7000` and accessible by the controller using the `webread` MATLAB function, which executed GET requests to hit the server. On initialization of the controller loop, the parameters for `MaxResponseTime`, `Complexity`, and `Duration` were set for each specific experiment, with the default values of 2 seconds, 8, and 75 seconds, respectively.

On initialization of the node microservice, the `socket.io` javascript library was used to establish and subscribe sockets to endpoint invocation events. The event callback handler for each endpoint would use these sockets to emit the configuration data to the Angular frontend.

To run the Node microservice, Node 10.16.3 or greater must be first installed. To install necessary dependencies, within the project directory, `npm install` must first be run. Finally, to serve the application on `localhost`, `node server.js` must be run.

4.1.2 Angular Web Frontend

The Stroop test web interface was developed using Angular, a frontend single page application framework with a single main component and a web service to communicate with the node microservice. The full source code and build files can be found on Github. The application is based on the research conducted by Delaney and Brodie, where the stress test is divided into four main phases: a color naming test for nonsense words, a mental math test, a traditional Stroop test, and a color naming test for words with common color associations.

The frontend web interface, on top of automating test start and stop periods, also includes metrics on response times for individual tests and correct response rates. These metrics are stored internally in the application and can be downloaded after each test is complete. To run the Angular frontend interface, Node 10.16.3 or greater and the Angular-CLI must be installed and added to the necessary environment path variables (this may have to be done as the root user). If the web application was cloned from the Github repository, the application must first be built in order to be served by the node microservice. This can be done by running the commands npm install and subsequently `ng build --prod` to build the application with the Angular CLI in production mode. Finally, the application is served by starting the node microservice with the aforementioned command `node server.js`. The application should then be live at the address localhost:7000, which can then be accessed via any modern web browser.

The application life cycle is controlled by a timer variable that represents the time to test start, which must be passed via the node microservice from the controller in MATLAB. As will be discussed in the

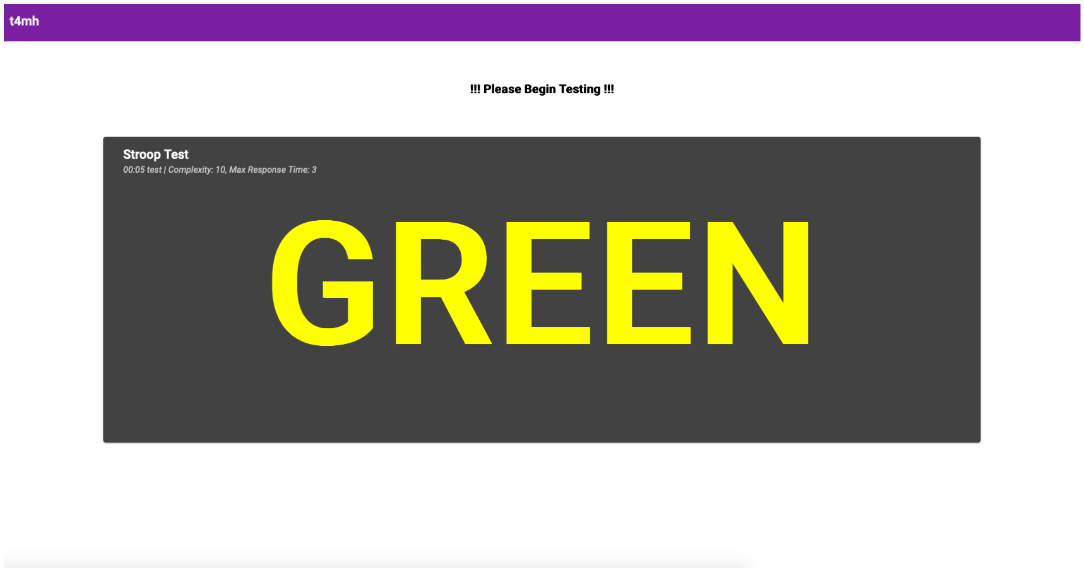

Figure 3: Angular web application.

following sections, the controller will, upon each iteration of the control loop, decrement this timer until it reaches zero, which will trigger the series of stress tests to start. Then, using the values defined by the variables `MaxReponseTime`, `Complexity`, and `Duration`, representing the maximum amount of time allowed for each question response, the number of different random colors used in the Stroop test, and each individual subtest duration, respectively, the tests will automatically begin.

4.2 Controller Development

The feedback controller was developed using MATLAB 2019b and has four main components: physiological data acquisition from Empatica E4 wearable sensors, HRV metric calculations, feedback control calculations, and finally music modulation. The code for the full feedback controller can be found on Github.

The control loop is normalized to update once every time period T, which was typically set to range between 0.2 - 1s depending on the experiment, meaning data collection from the sensor, HRV metric calculations, feedback controller update, and music modulation each occur once every T seconds. To ensure a consistent sample time is sustained, a manual delay is added to the controller logic during each loop iteration.

4.2.1 E4 Connection

The Empatica E4 wristband is a wearable sensor for collecting biosignals from participants for feedback control. In addition to providing HRV metrics from a PPG signal, the E4 wristwatches also include a galvanic skin response (GSR) sensor, a three-axis accelerometer, and a skin temperature sensor (see Table 2). The E4 wristwatch has multiple modes of data collection, for both online control via bluetooth low energy (BLE) streaming to a nearby computer or the E4 mobile application, or offline analysis by saving the recorded data locally.

Table 2: E4 Signals

Signal Name	Description	Sample Rate	Units
ACC	3-axis accelerometer sensor	32 Hz	-2g, 2g
BVP	Blood Volume Pulse, data from PPG	64 Hz	-
EDA	Electrodermal Activity (galvanic skin response)	4 Hz	μs
IBI	Inter-beat intervals (RR-int)	64 Hz, filtered	ms
TEMP	Skin Temperature	4 Hz	°C
HR	10-second window HR	-	bps

The E4 devices can be managed with the Empatica E4 Manager Software, which facilitates connecting to the E4 devices, updating firmware, and syncing any stored data files on the E4 devices. To facilitate real time connection to the E4 and data streaming, the E4 Streaming Server is required, which can also only be run on Windows machines. This streaming server can faciliate connections to up to five E4 devices simultaneously via BLE. The streaming server then uses by default port 28000 on 127.0.0.1 to establish a streaming server connection that can then be accessed in MATLAB.

After E4 devices have been connected to the Empatica streaming server, data can be streamed in real time in MATLAB with the `tcpip` function. Individual streams to each Empatica E4 device can then be established via the following commands (note that this process can also be performed using a client such as telnet for debug purposes):

1. `device_list`: lists all available device addresses
2. `device_connect <DEV_ID>`: connects to specified device address
3. `device_subscribe <STREAM_NAME> ON`: starts streaming data from specfic stream name (GSR, IBI, HR, ACC, TEMP, BVP)

After a streaming connection has been established to a particular E4 device, data can be read line by line from the TCPIP connection via the `fscanf` function in MATLAB. Data from the E4 devices is sent space-delimited with three columns: a tag name (in the form `E4_Ibi` or `E4_Bvp`, etc), a timestamp in ms, and a data column corresponding to a specific stream type. Upon completion of the controller script, all TCPIP connections are closed and destroyed.

4.2.2 HRV Metric Calculations

HRV metrics are calculated with two main methods: a heuristic method developed for online controller input, and an established method using Kubios HRV Standard for offline analysis and mhrv, an open source MATLAB HRV toolbox. The former, ultra-short-term metric relies on the pre-established window length of 120s or shorter, and focuses on calculating SDNN from RR-intervals.

First, a data calibration period is performed in order to obtain normal values for RR-intervals, which can be used during testing to exclude outlier values that often occur as an artifact of motion. Participants are asked to be as still as possible as a normal range for RR-intervals is obtained. This period is also used to calibrate the peak-finding algorithm that is used to find local minima in a participant's PPG signal, which is then ultimately transformed into RR-intervals.

Once calibration is complete, RR-intervals must be obtained, and are done so with a combination of two data feeds: IBI (inter-beat intervals) and BVP (blood pressure pulse) from the E4 devices. IBI is calculated by the E4 devices and takes advantage of motion artifact filtering that excludes data from the E4s during periods of extreme motion, which can be detected using the accelerometer senor integrated into the wristwatch. Values from IBI are preferred for RR-interval calculation, however they are not always available since they represent a filtered value. When IBI is not available, BVP is then used, representing a raw PPG signal from the participant. In this case, the MATLAB `findpeaks` function is used to generate RR-intervals.

Raw RR-values must further be processed to obtain valid SDNN metrics using the open source MATLAB toolbox mhrv (documentation can be found here). mhrv was created by researchers at PhysioZoo, an open source project for analzying HRV for human and animal elctrophysiological data [25]. Developed to work with the PhysioNet WFDB [26] data format, mhrv provides a variety of ECG, WFDB and RR-interval processing subroutines, however it was used specifically for three functions to detrend RR intervals, filter RR to NN, and calculate SDNN, respectively: `mhrv.rri.detrendrr`, `mhrv.rri.filtrr`, and `mhrv.hrv.hrv_time`. This process is illustrated in Figure 4.

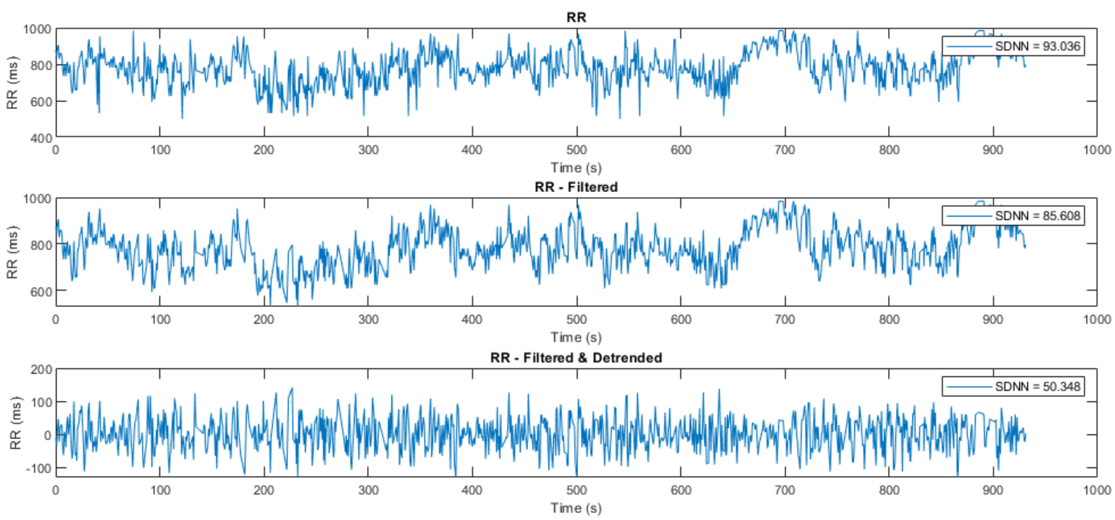

Figure 4: RR to NN generation.

When analyzing RR intervals for HRV metrics to measure stress, a stationary assumption of the RR-interval data is assumed whereby ultra-low and low frequency components of heart rate are assumed to be negligible. Since many of the feedback control experiments can range from 10 to 20 minutes, this assumption becomes no longer valid, and therefore RR interval data must be detrended to effectively analyze SDNN and other time domain metrics [27]. mhrv provides the function `mhrv.rri.detrendrr` which when applied to an RR signal, will generally lead to a lower SDNN value as the low frequency components will be removed.

Next, once a detrended RR signal has been obtained, NN intervals are calculated using the `mhrv.rri.filtrr` function, which applies three different outlier detection filters on raw RR data: a range-based detection, a moving-average filter based detection, and a quotient filter based detection. Range-based detection filters all RR intervals that fall outside of acceptable ranges for RR, 0.32 s (187.5 BPM) to 1.5 s (40 BPM). Next, moving-average filter-based detection removes values that fall outside of an acceptable delta from a moving average of previous RR-intervals, defaulting to a delta-threshold of 20% from the previous 10 samples. Finally, quotient filter detection removes RR interval values that exceed a maximum different from the previous sample, with a default threshold of 25% difference.

After detrending and filtering, NN intervals are then used to compute SDNN via the `mhrv.hrv.hrv_time` function, which calculates not only SDNN but also other time domain HRV metrics, including average NN duration, RMSSD, pNNx (the percentage of NN intervals which differ by at least a set number of milliseconds from their preceding intervals, with a default value of 50 ms difference), and standard error of mean NN interval length.

4.2.3 Feedback Control Design Approaches

4.2.3.1 RR Interval Control At its most fundamental level, HRV changes each time the heart emits a new beat. This event is captured directly through the RR interval signal, which updates immediately each time a new heart beat is emitted. Therefore, one of the first strategies for feedback control was through an attempt to directly influence the value of RR intervals and command a set amount of variability using a sinusoidal reference input (in ms):

$$r(t) = \frac{100}{HR_{10}} + A_{max}\sin(2\pi t f_{peak})$$

where HR_{10} is a 10 second windowed HR, A_{max} is chosen to be 80, representing a target SDNN value of 100 which is significantly higher than a baseline value of 48.5 [7], and f_{peak} is the peak frequency chosen to be 0.2 Hz, representing an HF band frequency. This signal is shown in Figure 5 as the orange sinusoidal signal, in comparison to the plant output shown in blue. Because of the offset term, the low frequency component of the sinusoidal reference signal tracks the low frequency component of the plant output.

4.2.3.2 SDNN Control While the literature suggests a minimum window length of 120 seconds for ultra-short term SDNN calculations, a wide range of SDNN window lengths were explored, ranging in the lower end from 10 seconds up to the recommended 120 second window. The primary reason for exploring SDNN window lengths less than the recommended 120 second window was to reduce the total amount of delay in the control loop, since longer window lengths will reduce the significance of more recently obtained RR-intervals. An important note to consider is that the case of directly using RR-intervals represents a lower bound on window length, and specifically corresponds to a window length of approximately 1 second (depending on RR interval values this can range between 0.32s to 1.5s).

4.2.3.3 PID Tuning A PID controller was chosen for the initial controller, often without the derivative term because of noisy sensor data. Because of its intuitiveness and performance potential, a PID controller provided the most promising initial results. Using the intuition gathered from the literature regarding the high level relationship between music features such as tempo, volume, and pitch and their corresponding effects on HRV and stress, a simple SISO controller was designed with an input from the current measured SDNN from a rolling window and an output ranging from -1 to 1 for music modulation.

4.2.4 Music Modulation

Three main features of music were explored to be modulated in the base controller loop: volume, pitch, and tempo. These are each modulated using functions native to MATLAB, however there should be exploration into using other technologies to provide more advanced music modulation. As for the choice

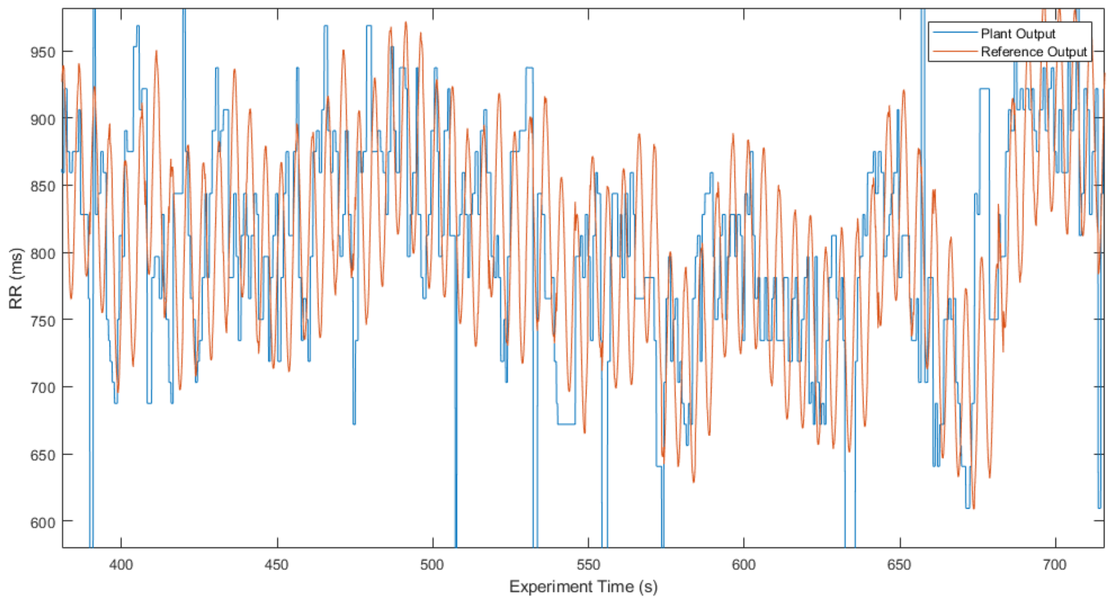

Figure 5: RR interval feedback control.

of a song, the 2011 song “Weightless” by Marconi Union was selected as the base song for modulation. “Weightless” has been commonly referred to as one of the “most relaxing songs in the world”, and while substantial research on its ability to quantitatively reduce stress levels has not been fully explored, it does possess several key qualities for music modulation, including a lack of lyrics, a generally ambient quality, and a consistent baseline tempo and volume throughout the song.

Volume is modulated using the `DampedVolumeController` object declared within the audiopluginexample. This object allows for the specification of a volume gain value in dB as well as a transition delay value in seconds. Modulating pitch and tempo is considerably more involved because of the close relationship between pitch and tempo. Tempo can easily be changed by resampling the audio file, however in doing so, with no other alterations, the pitch will also inherently be affected: speeding up a song causes pitch to rise correspondingly while similarly, slowing down a song causes pitch to drop correspondingly.

Changing tempo independently from pitch creates several unique problems with feedback control, since many of the algorithms developed for Time Scale Modification (TSM) in music are not meant to be used in real time, i.e. there exists significant delay due to the computational processing time of initializing a phase vocoder for an audio file, a popular frequency domain solution to TSM. This can be solved by reducing the length of audio being time stretched, effectively “chunking” an audio file into small, <1s audio segments. These pieces can then be individually time stretched to new tempos demanded by the controller.

MATLAB provides the `stretchAudio` function introduced in R2019b and was used as a wrapper for a phase vocoder based TSM. Simply segmenting an audio file into short segments and then running `stretchAudio` on each segment, however, introduces problems in the audio fidelity, specifically during the transitions between individual audio segments, where during playback an audible pop and pulsing sound can be observed.

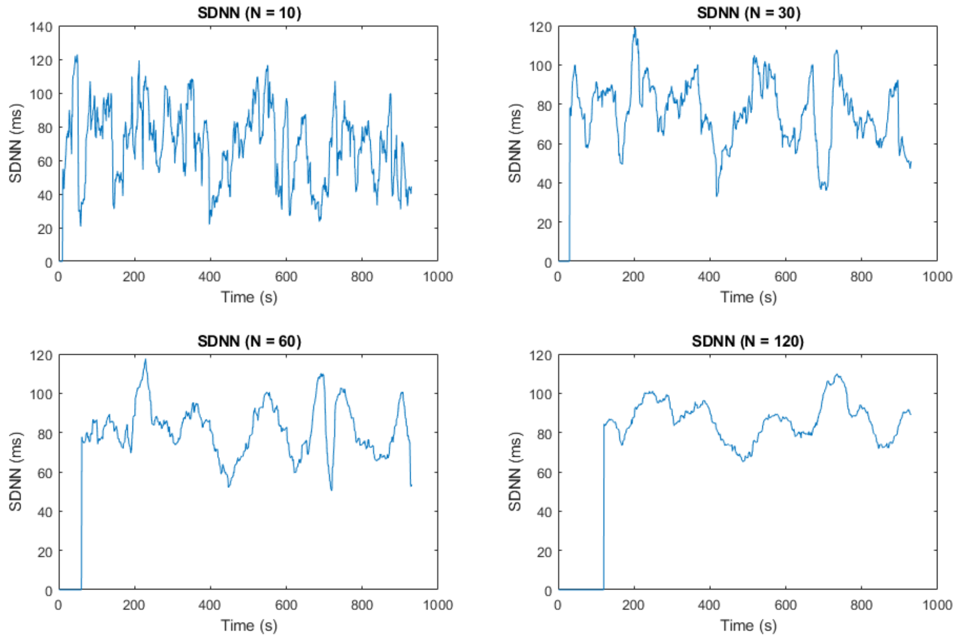

Figure 6: SDNN Window Lengths

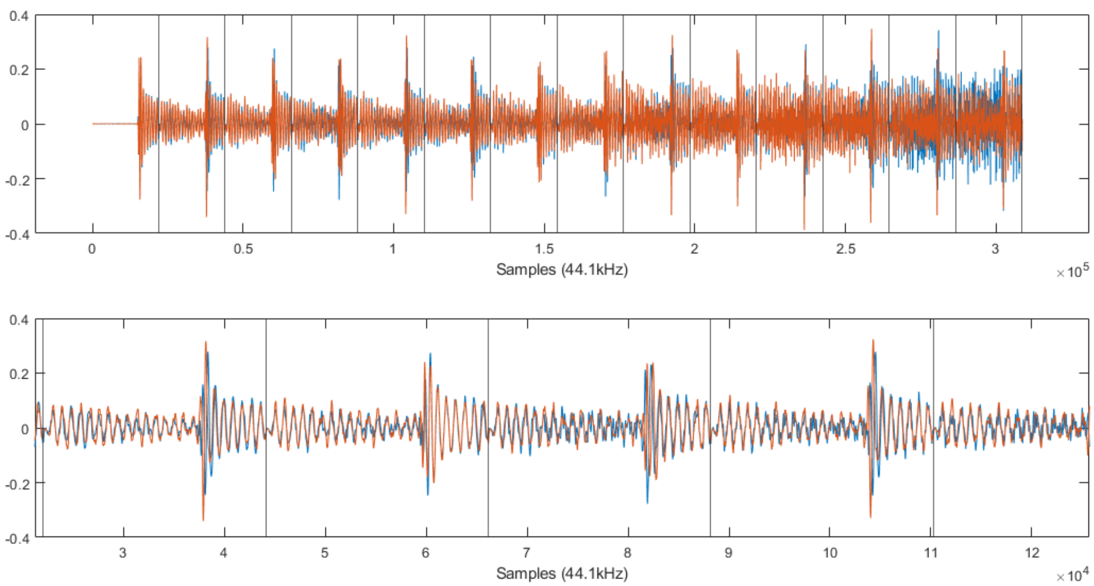

Figure 7: TSM issues.

Vertical lines represent breaks between subsequent audio segments, where the signal can be observed to taper.

Two main issues were observed. First, the `stretchAudio` function results in the tapering of the audio signal at the immediate start and end of the audio segment, leading to the pulsing sound observed during playback. Second, due to numerical issues and with subsequent audio segments not necessarily being modulated to the same tempo, discontinuities in the audio are created when time-stretched audio segments are reconstructed during the transitions between subsequent audio segments. To address the former problem, a buffer both before and after each audio segments, the length of which could be tuned for best performance but typically ranged between 10-50% of the original audio segment length, was included when time-stretching the audio segment. This start and end buffer was then discarded when reconstructing the audio signal, eliminating the tapering effect. To address the latter problem, subsequent audio signals were "mixed" together by fading the preceding audio signal out while fading the succeeding audio signal in. This is described below in the following algorithm:

Algorithm 1: Real-Time TSM

```
buffer = 0.5;                                                    // Set buffer to 50%
songPointer = 1;
while Song is not finished do
    nextSongChunkLength = tempo * (1 + buffer) * BASE_CHUNK_LENGTH;
    bufferLength = 0.5 * buffer * BASE_CHUNK_LENGTH;
    /* TSM using MATLAB stretchAudio algorithm                                      */
    audioSegment = song(songPointer : songPointer + nextSongChunkLength);
    rawModifiedSegment = stretchAudio(audioSegment, tempo);
    modifiedSegment =
     rawModifiedSegment(bufferLength:bufferLength+BASE_CHUNK_LENGTH);
    /* Smooth audio segment endss                                                   */
    for i ← 1 to bufferLength do
        modifiedSegment(i) = weights(i)*previousEndBuffer(i)+(1−weights(i))*modifiedSegment(i);
    end
end
```

5 Results

5.1 System Identification Experiments

One of the first steps to effectively designing a controller was to obtain both an intuition and a more quantifiable relationship between features in music and their effects on not only stress levels but specifically HRV metrics. For ease of analysis, tempo and TSM modulations were explored in most detail. Pitch modulations were not explored mainly because of the adverse effects of dynamically changing the pitch of music within the same song, since it introduces compositional differences in the music that may sound unsettling to a listener. Volume modulations, while potentially effective, were not explored because of the relative inconsistency in actual output volume, which depended not only on the playback gain set via the controller but also the system volume of the machine running the algorithm and the output gain on any sort of external speaker system.

To obtain this intuition and basic relationship, several open-loop feedback experiments were conducted in the hopes of finding an approximate first order response between tempo modulations and HRV metric changes (the plant, however, has been found to be highly nonlinear). First, a mapping between controller output, which ranged between -1 to 1 with saturation limits, and tempo was created by assigning a maximum and minimum tempo change factor. These values were tuned heuristically by allowing a participant to choose an upper bound, typically between 1.5-2 times the tempo, and a lower bound, typically between 0.5-0.8 times the tempo. Tempo modulation also had algorithmic limits in range, as extreme modulations would result in large amounts of degradation and corruption in audio quality. Finally, once the bounds on tempo were chosen, a second degree polynomial was fit between the two bounds and used to map the controller output to a tempo change value.

One of the first inputs that was passed to the system was a simple step input, or in some more sophisticated trials, a square wave input. These experiments would allow the participant to listen to a song at normal tempo for five minutes (the minimum interval for short-term HRV analysis) and then immediately increase the tempo to the maximum upper bound, in an attempt to shock the plant. The results from these experiments were generally difficult to interpret because of noise within the plant output, which could be caused by a variety of factors including sensor errors or unaccounted disturbances to the plant (i.e. stressful thoughts or physical discomfort during the experiment). Step input experiments, however, highlighted that SDNN did have some changes in the short term to the actual change in controller output, and lead to the development of the next plant inputs with triangle waves.

With the intuition that the change in controller output could have an observable response on the system, a set of experiments were designed to use triangle wave inputs with varying frequencies in order to potentially obtain information not only on plant response but also important metrics for creating a first order approximation such as plant delay. Six different triangle inputs were given to the plant, with period lengths ranging from 20s to 200s. A two minute buffer period was also included to ensure that a 120s SDNN window was valid. Music tempo was modulated with the input signal at a period of 1s, and a 120s SDNN window was calculated with each tempo iteration. Figure 8 overviews these results, which generally support the literature that slower tempos, or more specifically, changes to create slower tempos, increase SDNN values which correspond to an decrease in stress levels, while faster tempos and changes to create faster tempos lead to lower SDNN values which correspond to an increase in stress levels. The impacts of these findings are most prevalent with longer period lengths, such as those of the 180s to 200s triangle periods, where SDNN exhibits the largest change, suggesting that high frequency changes are being filtered out by the plant to an extent. Finally, these figures point to some intuition on delay present in the system, which appears to be approximately 20-40s. Figure 9 provides an extended look into the longer triangle period lengths, where all triangle wave inputs had a period of 200s. All other experimental parameters, such as SDNN length and a pre-input buffer period, remain consistent with those in Figure 8.

The intuition gathered from visually observing the input output relationship with tempo changes can also be furthered confirmed by an analysis into the crosscorrelation between the audio signal provided to the plant and the NN intervals obtained from HRV metrics. By incorporating the tempo changes into the music audio file being played, the time delay between the input audio and NN signal can be estimated with the peak crosscorrelation in Figure 10, which occurring at 35.48s corresponds with the visual results. It is important to note however that crosscorrelation is also relatively high for delays up to approximately 200s, suggesting that the true delay between tempo changes and HRV metrics is actually variable and may depend on factors such as the type of music being played.

5.2 Controller Performance

With the intuition gathered from the system identification experiments that changes to slower tempos tend to increase SDNN while changes to faster tempos tend to decrease SDNN, as well as the general range of SDNN changes and approximate delay, a first order model could be approximated for this plant and subsequently PI controller gains can be estimated. Ziegler and Nichols presented a design methodology for PID controller gains in 1942 that utilizes an open loop step response parameterized by two variables, a and L. After a step response has been obtained, a tangent line is drawn to the step response where the

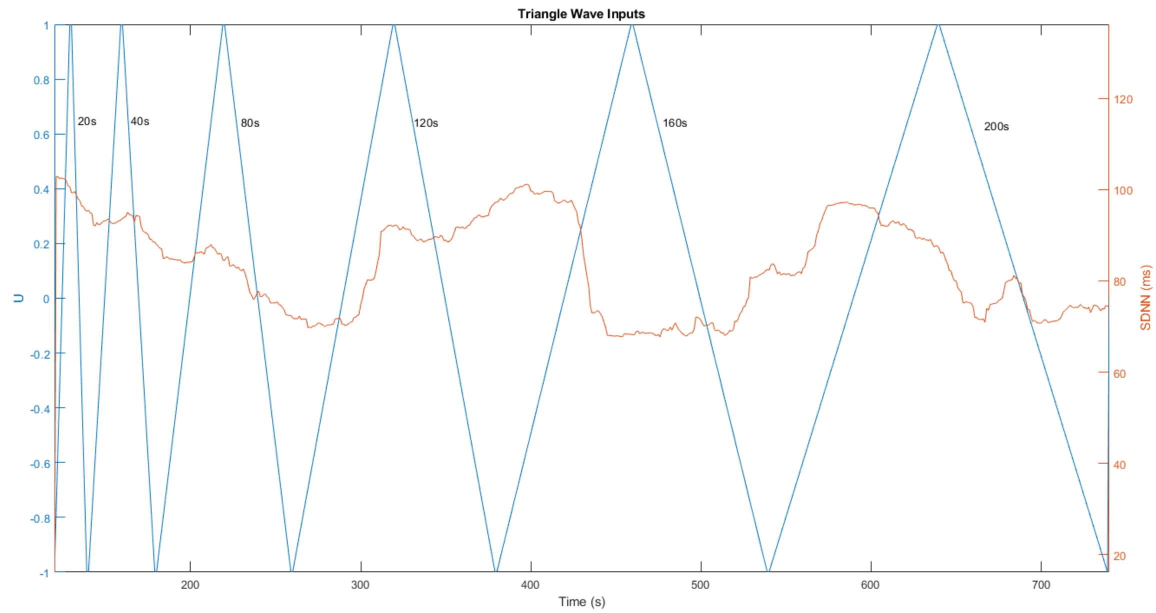

Figure 8: Plant output with variable triangle signal tempo input.

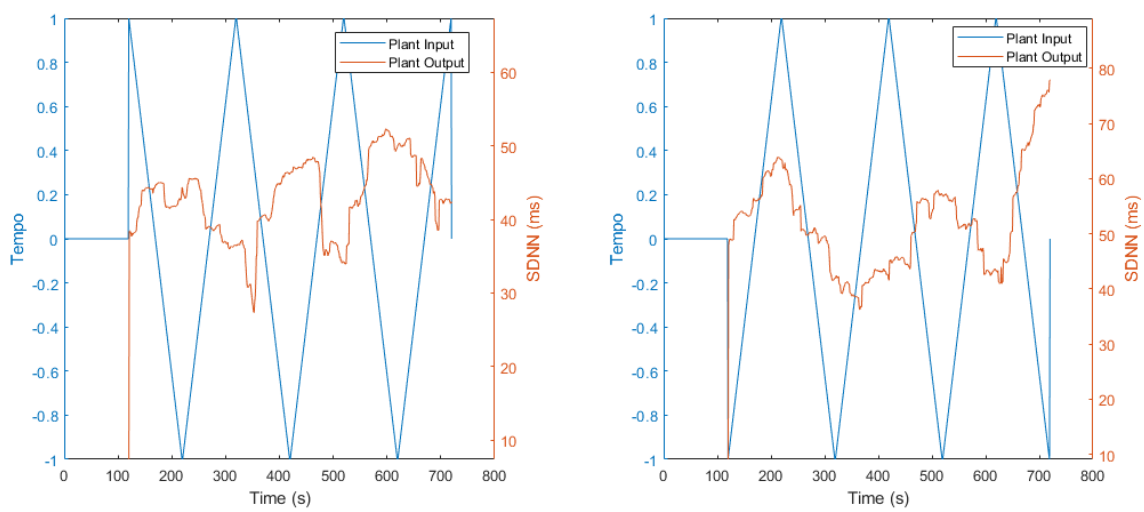

Figure 9: Plant output with triangle signal tempo input.

slope reaches its maximum value, and the intersection between this tangent and the y-axis provides the parameters a and L, where a represents the y-intercept and L represents the x-intercept. The PID gains can then be calculated using the equations listed in table 3 [28].

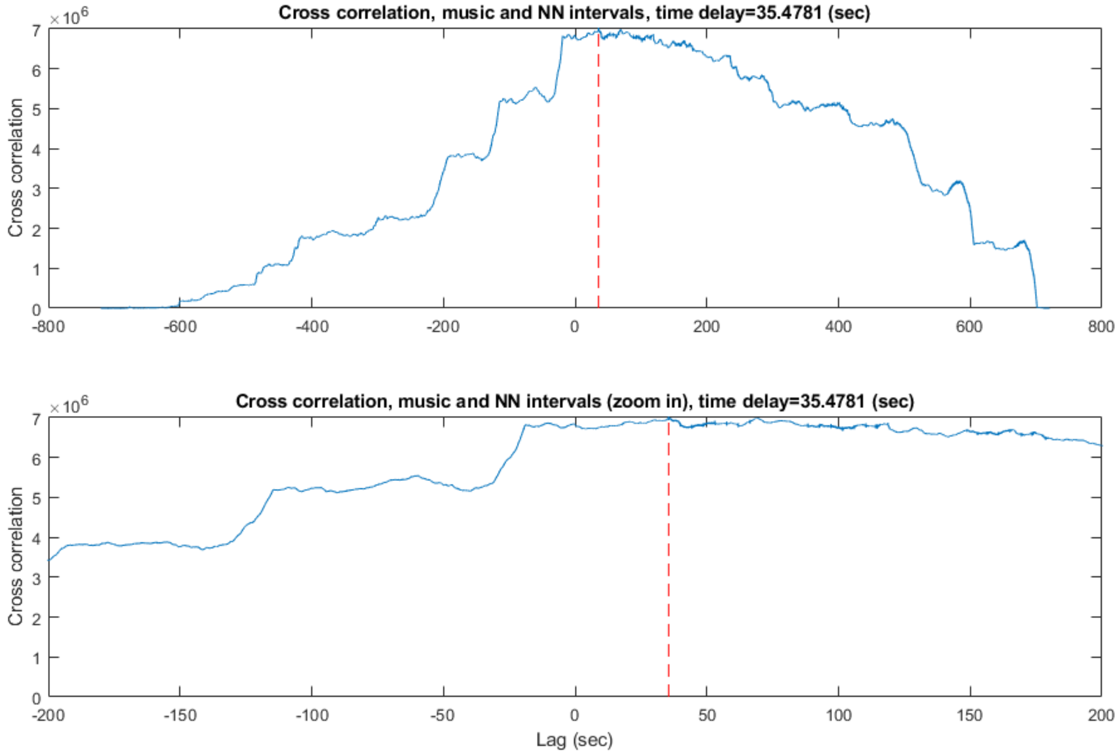

Figure 10: Plant input and output crosscorrelation.

Table 3: PID Controller Parameters obtained from Ziegler-Nichols step response method

Controller	K	T_i	T_d	T_p
P	$1/a$			$4L$
PI	$0.9/a$	$3L$		$5.7L$
PID	$1.2/a$	$2L$	$L/2$	$3.4L$

Using this method, a was estimated to be 30 and L, corresponding to a delay, was estimated to be 30. These values were subsequently tuned manually to ensure that the control signal output would generally avoid saturation by increasing a to 60 and L to 50, which resulted in a smaller value for K_p and a larger value for T_i. To explore the effects of varying window lengths and potentially reducing loop delay, multiple SDNN window lengths as inputs were explored, ranging from 10s to 120s. The results for $SDNN_{30}$ and $SDNN_{120}$ are compared, representing an SDNN input using a 30s and 120s window, respectively. Additionally, a feedback controller designed using RR-intervals as a control input is included in this analysis.

By the design of evaluation experiments which use a modified Stroop stress test as "disturbance" input to the system, the goal of any controller for these experiments and thereby a marker for success is the ability to sustain a baseline SDNN value through the disturbance input of the system. This indicates that a reference value for SDNN to be used by the controller should be chosen using a five-minute pre-test calibration period that allows for the controller to have a more relevant reference input. To assist in potentially increasing SDNN through the stress test, reference inputs were often chosen to be a factor slightly above the recorded baseline value. Figures 11 and 12 illustrate the performance of the various controllers previously discussed, as well as offers a comparison between two control cases: SDNN during a test with no music, and SDNN during a test with music, but without any modulation (no controller). These two control cases offer insight into how much improvement using the controller can potentially bring to sustaining SDNN values, and ensures that any potential benefit is not merely from the introduction of music itself but rather the modulation of music.

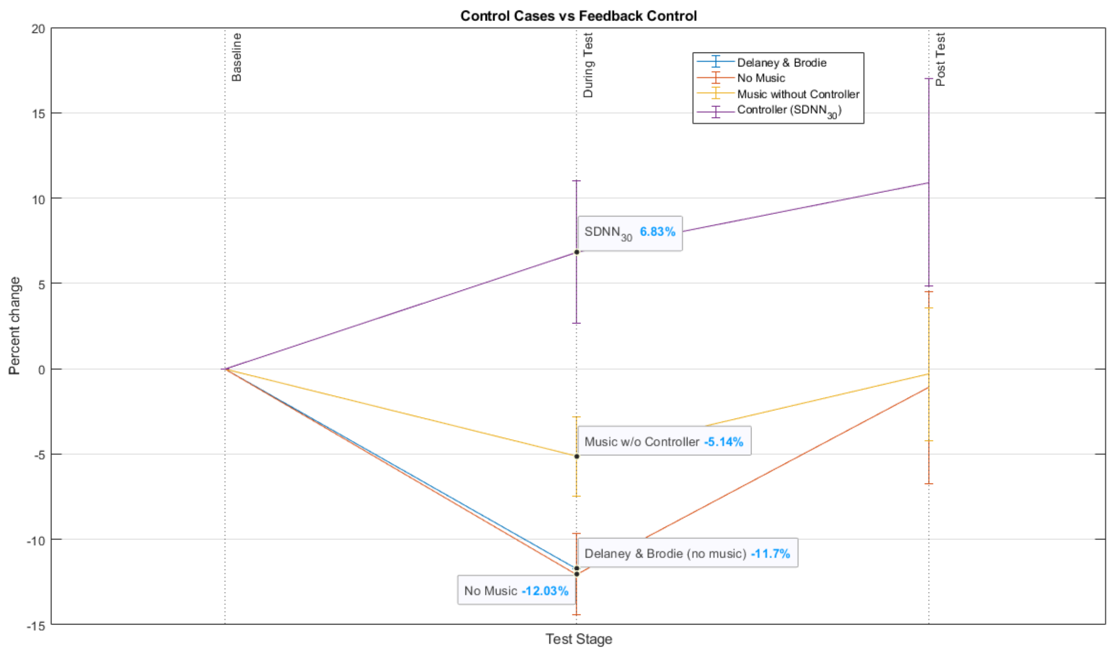

Figure 11: Comparison of control cases to use of controller.
Note that Delaney and Brodie does not include a post test recording of SDNN with their results and is therefore omitted from this figure.

The values obtained from these experiments are both consistent with the results from Delaney and Brodie, who using the same stress test obtained comparable results to the case of no music, but also with the expected baseline values for SDNN for subjects within this particular age and demographic range (men 20-29 years old, median SDNN of 48.5) [7]. Ultimately, these results suggest that with a properly designed and tuned controller and a strategically chosen window length for SDNN input, feedback control and regulation of stress levels can be potentially successfully obtained for an individual. While ultra-short term SDNN metrics often do not cross the lower bound of the two-minute SDNN metric, the use of a shorter SDNN window is primarily motivated by the fact that shorter window lengths, such as 30s, provide significantly less delay to the current state of the plant, and therefore have greater potential in terms of regulating SDNN long term. Additionally, shorter window lengths exhibit greater amounts of total variation, and may actually be beneficial to the overall performance of the controller as previous experiments have demonstrated that a change in tempo can drive changes in SDNN.

A general note to include about the data presented in this plots is that it is extremely limited in sampling across a diverse and large population. N = 6 for $SDNN_{30}$, N = 8 for $SDNN_{120}$, N = 2 for $SDNN_{RR}$, N =

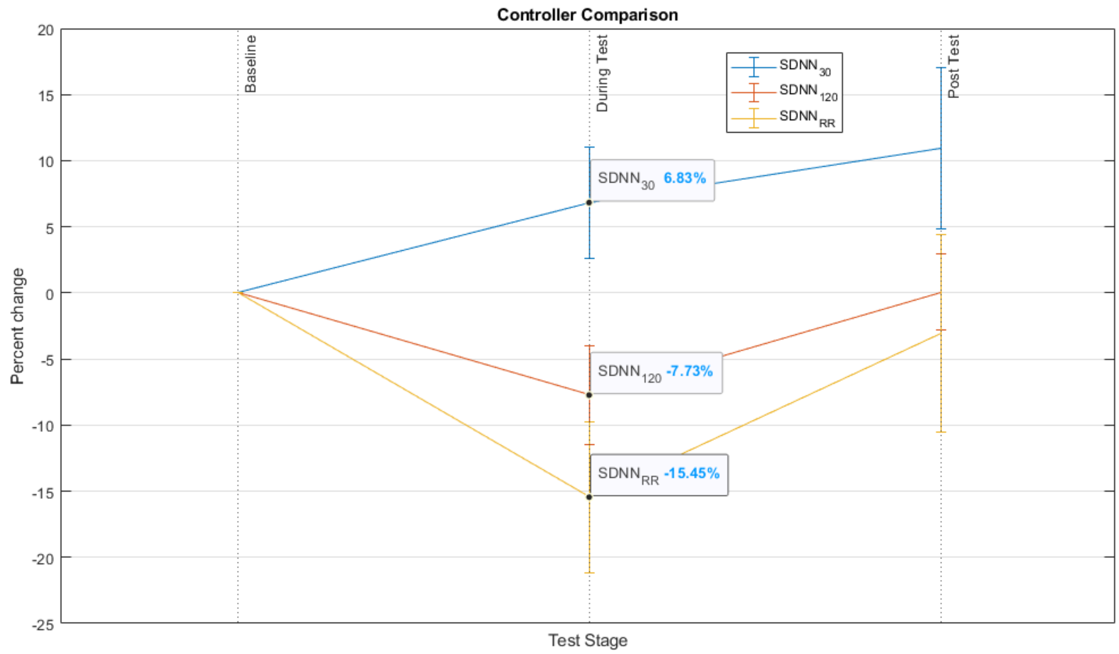

Figure 12: Comparison of various controller performances.

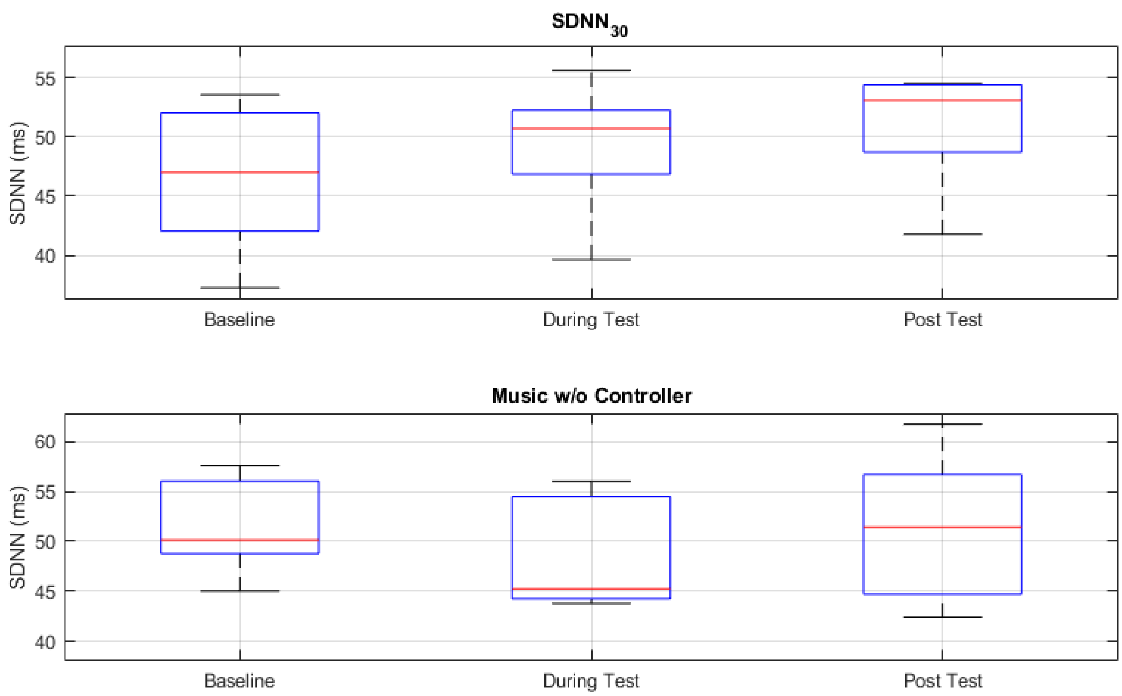

Figure 13: SDNN changes.

7 for music without control, and N = 4 for no music. Specifically, while the performance for $SDNN_{RR}$ is significantly worse than the other cases, it represents only two samples and could be improved with better tuning and controller design. These results should not be used to rule out particular approaches for their long term efficacy.

6 Conclusion

6.1 Future Work

While the ultimate goal of this thesis was to design a feedback controller to regulate stress, the ultimate goal of this research as a whole is significantly more ambitious: to design a robust control algorithm that can, with minimal amounts of calibration, actively reduce stress levels of not a single individual but any arbitrary individual with music modulation. This goal introduces what is commonly known as robustness in controls: while it may be possible to manually tune a single controller to work with a single plant, it is significantly more difficult to design a robust controller that is able to handle an entire class of plants. This goal will ultimately require much more extensive testing beyond the limitations of this thesis to a handful of volunteer participants.

Before controller design advances further, however, there are yet still several fundamental issues and assumptions that must be addressed for long term success. First, while SDNN and other HRV metrics are generally accepted in literature for their ability to provide a metric for stress levels, the large body of research conducted focuses on high level trends: higher HRV levels are associated with relaxation, while lower are associated with stress. Granular analysis of HRV metrics, especially for ultra-short term metrics that are required when conducting real-time feedback control, has yet to be extensively researched. A true, quantifiable metric for instantaneous stress levels has yet to be firmly established. The SWELL-KW workplace stress dataset provides an initial starting point for this analysis, as it provides highly granular physiological, body posture, and facial expression data that is also tagged with the onset of stressors [29]. This thesis has additionally assumed in general some degree of linearity in terms of the relationship between HRV and stress, however this assumption can easily be shown to be false.

Next, the issue of music modulation must be addressed. Changing features such as tempo, volume, and pitch may not be the most effective way to influence an individual's stress levels; this thesis chose these parameters primarily because of their relative ease in terms of algorithmic modulation with MATLAB. Another particular strategy for control is to significantly reduce the sample time of the controller to the range of minutes; this would effectively allow the controller to act as an affective music player, choosing specific songs for their potential to reduce stress rather than modulating individual songs [18]. A second approach is considerably more radical, and involves abandoning familiar music altogether and enters the world of algorithmic composition. Generative music, or music that is created via a set of rules and conditions with added randomness, is a genre of music that has not been explored extensively in research however has been pursued by many composers, most notably Brian Eno and Philip Glass, the former of whom has released entire albums of generative music. Generative music allows for even more granular control of the plant input, since it is created entirely from tunable parameters and rulesets that can be potentially modulated using a feedback controller. Data collected from human subjects testing at the Schoenbaum Family Center includes a preliminary experiment into the effects of various parameters of generative music on the stress of human subjects, in particular young children. This dataset provides HR and other physiological signals as well as information on audio inputs and can be used to potentially explore both generative and traditional music modulation approaches. It is accessible through Dr. Hugo Gonzalez Villasanti.

6.2 Acknowledgements and Contributions

I would like to first acknowledge my research advisor, Professor Kevin Passino, for his continued support throughout the entire duration of the preparation of this thesis, from introducing me to this problem and his research group, providing insight and direction, and constant feedback and expertise in the controls field. In addition to Prof. Passino, I would like to thank Dr. Hugo Gonzalez Villasanti who I had the fortune of learning from and working with both while he lead research on stress with our group and as he transitioned to focusing with the Schoenbaum Family Center (SFC), where he helped immensely in collecting data from human subjects. On that note, I'd also like to acknowledge all of the researchers and staff at SFC for facilitating conducting human subjects testing and for their invaluable help to prepare an IRB. I thank Prof. Phil Schniter and Dr. Laura Justice for serving on my defense committee and offering their invaluable feedback. Finally, I want to thank my family for always supporting my endeavours, and my friends, for both their support and for occasionally serving as another subject to test on.

References

[1] Castaldo R, Montesinos L, Wan TS, Serban A, Massaro S, Pecchia L. Heart rate variability analysis and performance during a repeated mental workload task. In: EMBEC & NBC 2017: Springer; 2017. p. 69–72.

[2] Teisala T, Mutikainen S, Tolvanen A, Rottensteiner M, Leskinen T, Kaprio J, et al. Associations of physical activity, fitness, and body composition with heart rate variability–based indicators of stress and recovery on workdays: a cross-sectional study. Journal of Occupational Medicine and Toxicology. 2014;9:16.

[3] Teisala T, Mutikainen S, Tolvanen A, Rottensteiner M, Leskinen T, Kaprio J, et al. Associations of physical activity, fitness, and body composition with heart rate variability–based indicators of stress and recovery on workdays: a cross-sectional study. Journal of Occupational Medicine and Toxicology. 2014;9:16.

[4] Yang Z, Jia W, Liu G, Sun M. Quantifying mental arousal levels in daily living using additional heart rate. Biomedical Signal Processing and Control. 2017; 33:368–78.

[5] McCraty R, Shaffer F. Heart rate variability: new perspectives on physiological mechanisms, assessment of self-regulatory capacity, and health risk. Glob Adv Health Med (2015) 4:46-61. doi:10.7453/gahmj.2014.073

[6] Goldberger AL. Is the normal heartbeat chaotic or homeostatic? News Physiol Sci (1991) 6:87-91

[7] van den Berg ME, Rjinbeek PR. Niemeijer MN, Hofman A, van Herpen G, Bots ML, Hilege H, Swenee CA, Ejgelsheim M, Stricker BH and Kors JA. Normal Values of Corrected Heart-Rate Variability in 10-Second Electrocardiagrams for All Ages. Front Physiol (2018). 9:424. doi:10.3389/fphys.2018.00424

[8] McCorry LK. Physiology of the autonomic nervous system. Am J Pharm Educ (2007). 71(4):78.

[9] Shaffer F, McCraty R, & Zerr CL. A healthy heart is not a metronome: an integrative review of the heart's anatomy and heart rate variability. Front Phychol (2014) 5:1040. doi:10.3389/fpsyg.2014.01040

[10] Thoma MV, La Marca R, Brönnimann R, Finkel L, Ehlert U, et al. (2013) The Effect of Music on the Human Stress Response. PLoS ONE 8(8):e70156. doi:10.1371/journal.pone.0070156

[11] Koelsch, S. et al. The impact of acute stress on hormones and cytokines, and how their recovery is affected by music-evoked positive mood. Sci. Rep.6, 23008; doi: 10.1038/srep23008 (2016).

[12] Jiang, J., Rickson, D., & Jiang, C. (2016). The mechanism of music for reducing psychological stress: Music preference as a mediator. The Arts in Psychotherapy, 48, 62–68.

[13] Fisher, S., & Greenberg, R. P. (1972). Selective effects upon women of exciting andcalm music. Perceptual and Motor Skills, 34(3), 987–990.

[14] Gan, S. K.-E., Lim, K. M.-J., & Haw, Y.-X. (2015). The relaxation effects of stimulativeand sedative music on mathematics anxiety: A perception to physiology model.Psychology of Music, http://dx.doi.org/10.1177/0305735615590430

[15] Iwanaga, M., Kobayashi, A., & Kawasaki, C. (2005). Heart rate variability with repet-itive exposure to music. Biological Psychology, 70(1), 61–66.

[16] Sandstrom, G. M., & Russo, F. A. (2010). Music hath charms: The effects of valenceand arousal on recovery following an acute stressor. Music and Medicine, 2(3),137–143.

[17] Patrik N. Juslin, John A. Sloboda, 15 - Music and Emotion, Editor(s): Diana Deutsch, The Psychology of Music (Third Edition), Academic Press, 2013, Pages 583-645, https://doi.org/10.1016/B978-0-12-381460-9.00015-8.

[18] Chiu, M., Ko, L. Develop a personalized intelligent music selection system based on heart rate variability and machine learning. Multimed Tools Appl 76, 15607–15639 (2017). https://doi.org/10.1007/s11042-016-3860-x

[19] Force T.: "Heart rate variability guidelines: standards of measurement, physiological interpretation, and clinical use", *Eur. Heart J,*, 1996, **17**, pp. 354-381

[20] Castaldo, R et al. "Ultra-short term HRV features as surrogates of short term HRV: a case study on mental stress detection in real life." *BMC medical informatics and decision making* vol. 19,1 12. 17 Jan. 2019, doi:10.1186/s12911-019-0742-y

[21] Castaldo R, Melillo P, Bracale U, Caserta M, Triassi M, Pecchia L. "Acute mental stress assessment via short term HRV analysis in healthy adults: A systematic review with meta-analysis." *Biomedical Signal Processing and Control.* 2015; 18:370–7.

[22] Delaney J, Brodie D. "Effects of short-term psychological stress on the time and frequency domains of heart-rate variability." *Percept Mot Skills.* 2000;91: 515–24.

[23] Munoz ML, van Roon A, Riese H, Thio C, Oostenbroek E, Westrik I, et al. (2015) "Validity of (Ultra-)Short Recordings for Heart Rate Variability Measurements." PLoS ONE 10(9): e0138921. https://doi.org/10.1371/journal.pone.0138921

[24] Stroop, J. R. (1935). "Studies of interference in serial verbal reactions." *Journal of Experimental Psychology*, 18(6), 643–662. https://doi.org/10.1037/h0054651

[25] Joachim A. Behar†, Aviv A. Rosenberg†, Ido Weiser-Bitoun, Ori Shemla, Alexandra Alexandrovich, Evgene Konyukhov, Yael Yaniv. 2018. "PhysioZoo: a novel open access platform for heart rate variability analysis of mammalian electrocardiographic data." Frontiers in Physiology 9 (2018): 1390.

[26] Goldberger, A. L. et al. (2000) 'PhysioBank, PhysioToolkit, and PhysioNet', Circulation, 101(23), pp. E215-20.

[27] Tarvainen, Mika P., Perttu O. Ranta-Aho, and Pasi A. Karjalainen. "An advanced detrending method with application to HRV analysis." IEEE Transactions on Biomedical Engineering 49.2 (2002): 172-175.

[28] Astrom K. J. Åström and T. Hägglund, PID Controllers: Theory, Design, and Tuning, Instrument Society of America, North Carolina, 1995.

[29] Koldijk, S., Sappelli, M., Verberne, S., Neerincx, M., Kraaij, W. (2014). The SWELL Knowledge Work Dataset for Stress and User Modeling Research. To appear in: Proceedings of the 16th ACM International Conference on Multimodal Interaction (ICMI 2014) (Istanbul, Turkey, 12-16 November 2014).