THE UNIVERSITY OF ADELAIDE
AUSTRALIA

Department of Mechanical Engineering
Final year Project 2004

Design and build a self-tuning stringed instrument

Andrew Leonard a1078401

Colin Love a1068091

Supervisor:
Frank Wörnle

Abstract

This report covers the research and development undertaken into the construction of a self tuning stringed instrument. This project is based around a conventional six stringed electric guitar.

Most guitars need regular retuning due to changes in temperature, humidity and string condition. The process of retuning, although relatively straightforward, can become tedious and repetitive. In the absence of an electronic tuner, tuning ‘by ear’ can only be achieved by an experienced musician, and often requires a fair amount of time and concentration. This project aims to provide a quick and automatic method of retuning. Such a system would greatly benefit live performers by reducing the time wasted on stage due to necessary retuning.

This self tuning system can be split up into three main components: the acquisition of guitar tuning, the electronic control system and the mechanical system. This entire system has to be able to maintain the tuning of the guitar when there is no power supplied to the system. Additionally the system should be able to provide the user with the option to tune the guitar manually if, for example, there is no suitable power source available. The system to be designed should not affect the aesthetics of the guitar; neither should it decrease its structural strength.

The acquisition system will include a means by which to detect string vibration, a filter system to separate the fundamental frequency of each string and a system to convert the signals to a usable format for use with a microcontroller.

The controller will then determine if, and by how much, each string needs re-adjusting and controls the motors in the mechanical system to achieve the correct pitch for each string.

It is proposed to have a mechanical system which uses motors to increase and decrease string tension to achieve correct pitch. This is to be achieved by having a motor spin a

threaded shaft, which in turn makes a threaded sleeve translate along its length. This is connected to a lever arm, which tightens or loosens the guitar string.

A one string prototype was constructed. This prototype formed the basis for the construction of a final complete system. This system has been demonstrated to achieve the desired performance specifications, whilst maintaining the functionality and aesthetics of the guitar, in which the control system is housed.

There are other products such as these on the market; however, these products are often more than five times the price of a high quality guitar. This project aims to achieve the same results at a much more affordable cost, so that this technology is available to the wider community.

Acknowledgements:

Many thanks go to Dr Frank Wörnle who has supervised this project throughout the year. Frank has been a great source of encouragement and has helped guide this project to its successful completion.

Thanks also go to Mr Jim Deer who spent many hours helping in the construction of the mechanical components.

We are also appreciative to Fretco for their helpful advice regarding guitar building techniques and materials.

Finally, our appreciation goes to the judges of the 2004 University of Adelaide Mechanical Engineering Project Exhibition who voted this project as the project with the Vipac prize for “Best Acoustics/Vibration Related Project”, and to Vipac for sponsoring this award.

Contents

Table of Figures

1 Introduction

1.1 Significance

Most instruments need constant retuning after having been played, left alone or after assembly or reassembly. This can be somewhat of a nuisance but usually cannot be avoided. For stringed instruments in particular, usually after a few songs have been played, some, if not all of the strings are slightly out of tune. Identifying and re-tuning these is usually a simple task, but can become tedious and repetitive after a while.

This problem has been overcome with the aid of modern technology such as electronic tuners, which electrically determines what pitch each note is being played (individually) and outputs whether the note is sharp or flat. However, this relies on the user to tune it manually, and concentrate on tuning. Many performers rely of having one or more spare instruments and someone dedicated to keeping them in tune, so that at any stage during a performance a recently tuned instrument can be requested. This takes less concentration on the part of the performer, so they are free to interact with their audience. However, this method is much more expensive than the former mentioned method, as it requires multiple instruments and a hired stage hand.

There is not much equipment on the market today that boasts a completely automatic self tuning system, and almost none reasonably priced and readily available. At the present time, only one commercial system is on the market, with others preparing for their release. These systems (to go into an electric guitar) are relatively expensive; approximately five times the price of a high quality guitar. This price does not include the guitar, which the customer has to supply to be fitted.

1.2 Aim

It is the endeavor of this research project to construct a stringed instrument that can retune itself without any external control. This project will be based around a conventional 6 stringed electric guitar. This would be of benefit to musicians in the live music industry (particularly those who are relatively inexperienced) who require frequent retuning of their guitars. This would give performers the opportunity for better crowd interaction and a more seamless performance, by reducing the time required to manually tune each string individually (and the concentration required if tuning ‘by ear’). Additionally, the aim of this project is to provide this functionality at a much lower price than existing models. Ideally, this system will not degrade the sound quality or aesthetics of the guitar, or significantly alter the size, weight or functionality of the guitar.

2 Problem Specification

2.1 Functional Specifications

To reach the target market of live musicians, the instrument must enhance the quality of the performance and/or make performing easier. To achieve this, a list of possible functions was derived.

- **Tuning Accuracy**

For the system to be at all worthwhile, it must be able to tune the guitar to an accuracy comparable with (or better than) that obtained by a competent musician. If the tuning system has perceivable tuning errors, it will not have any use.

- **Manual Tuning**

It is desired that the instrument retain the option of being manually tuned, to cater for circumstances when power is unavailable, or an alternate tuning is called for.

- **Sustainable Tuning**

It is required that the tuning of the guitar is not lost when power is removed from the system.

- **Tuning Speed**

The system must be able to tune itself with sufficient speed so as not to hinder live performances.

2.2 Physical Specifications

- **Size/Weight Limitations**

It is desirable to have a tuning system that fits within the body of the guitar. This negates the need for any changes to existing guitar cases, straps, leads and stands, and will not change the technique required to play the guitar. The weight of the guitar must also not be significantly altered.

- **Appearance**

As image is an important commodity in the target market, it is desirable that the appearance of the guitar is not significantly different from a conventional electric guitar.

2.3 Performance Specifications

To simplify the design, the system is only required to fine tune a guitar which already is relatively close to standard tuning. This limitation reduces the amount of string elongation that needs to be accommodated, and simplifies the filtering system required to separate string frequencies. It is desired to achieve a tuning range of plus and minus three semitones of each set frequency.

For the system to be useful in live performances, a tuning time of less than ten seconds is required. To achieve a tuning accuracy comparable with standard electronic tuners it is desirable to achieve an accuracy of five percent of a semitone.

3 Preliminary Research

3.1 Transperformance Guitar

The Transperformance guitar is the most highly advertised self-tuning system currently on the market. It boasts fast and highly accurate tuning to a vast array of tuning schemes. The high functionality of the system stems from its use of 2 main control systems and complex string modeling algorithms. The strings are tuned using a closed loop tuning system. Once in tune, alternate tunings can be achieved in an open loop control system by calculating the required tensions (and hence, actuator positions). This means that the user is not required to strum the strings whilst changing between alternate tunings. The system costs nearly US$4000 to be retrofitted to a guitar, and when combined with the cost of the guitar itself, is well beyond the budget of average musicians.

3.2 Rochester Institute of Technology

A student group from the Rochester Institute of Technology designed and built a self tuning guitar system in the US academic year ending in 2003. They attempted to implement a FFT algorithm to determine each of the string frequencies, however the algorithm did not determine all six frequencies simultaneously, and frequency determination was not concurrent with motor operation. These problems led to very slow response, and ultimately a "very unreliable system" as stated by their supervisor, Dr Dorin Patru[1]. Their tensioning system comprises of pre-existing tuning pegs coupled to DC motors, installed in the body of the guitar, behind the bridge. This method is simple, cheap and easy to prepare, as all components are off the shelf with minimal modification.

[1] Dr Dorin Patru is an assistant Professor in the department of Electrical Engineering at the Rochester Institute of Technology. www.rit.edu

4 Initial Concept

The design was split into three main categories;

- Mechanical system: to tension the strings for correct pitch
- Data acquisition system: to retrieve frequency of the strings
- Control system: to drive actuators to gain correct pitch of the strings

Shown in Figure 4-1 is a simplified block diagram of the various parts of the system to be designed. The arrows represent electronic/mechanical signal paths.

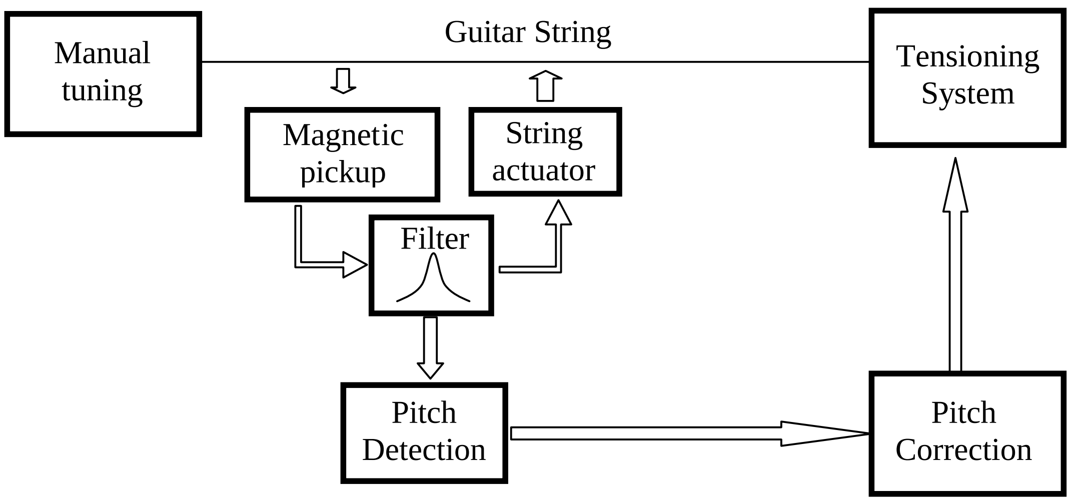


Figure 4-1 Flow diagram of entire system

In the flow diagram, it shows that the signal from the string is generated in the magnetic pickup. This signal is then refined using a bandpass filter. The frequency (or pitch) of this signal is determined and compared to a desired frequency, and from this comparison a control signal is generated and applied to a tensioning system. This process is part of a continuous closed loop system involving the instantaneous vibration of the string. The string actuator is a system which vibrates the strings automatically. This can be achieved either mechanically, or electromagnetically. With such an actuator, the entire system would be fully automatic.

4.1 Layout

Various options exist in regards to the layout of the system that was designed; the system could be external with motors arranged in a unit to be attached to the head of the guitar to make use of the existing tuning pegs. Although this is a non-destructive improvement to the guitar, it adds weight, and alters the visual aesthetics of the guitar.

A second solution is the use of a solitary motor, placed manually by the user over each tuning peg in turn. When the respective string is plucked, the motor tunes the string, with this process being repeated for each string. This process is time consuming, and therefore is of little advantage to the user over basic manual tuning.

A third alternative is to mount the system within the body of the guitar, therefore allowing both manual and automatic tuning of the guitar. This method requires significant modification of the guitar body, but remains less conspicuous than an externally mounted system, whilst allowing simultaneous tuning of all six strings.

5 Mechanical System

5.1 Specifications

The function of the tensioning system is the translation of a voltage into a change in the tension of the string, whilst maintaining the tension of the string when no power is supplied to the system. The maximum tension the tensioning system is required to endure is 117 N. This figure was derived, as shown in appendix A. The mechanical system must be sufficiently compact so as to fit discreetly within the body of the guitar, without affecting it structural strength. The strings on a guitar are spaced at 1cm intervals, requiring each separate tensioning system to be compact enough to fit within this constraint.

5.2 Bridge and Nut

The theoretical model of the relationship between string tension and frequency is a simplification of the practical behavior of guitar strings (appendix A, eq. 5). Much of the discrepancy of this model is related to non-ideal supports of the strings. On low to medium cost guitars, the string is supported by a plastic or bone ‘nut’ at the head end of the guitar and a metal ‘bridge’ at the body end of the guitar. The static friction at these points causes the portions of the string outside the supports to have tensions not consistent with the main vibrating portion of the string. This causes the most pronounced problems when a string is played heavily after being tuned. When played heavily, the tension difference at the supports is only held by kinetic friction, and hence the string moves closer to equilibrium, changing the pitch of the vibrating section of the string. This behavior is difficult to design a control system for, and hence it is advantageous to utilize low friction supports. A common solution to the friction at the nut is to install a nut made of graphite. On more expensive guitars, or those with ‘tremolo tailpieces’, the bridge is replaced by a set of 6 roller bearings seated in movable supports to lower the friction. A similar set of bearings can be used in place of the nut. A picture of a roller nut and roller bridge is shown in Figure 5-1.

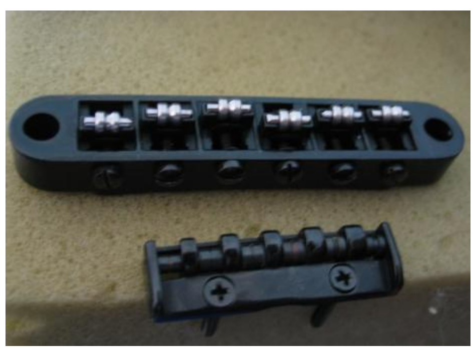


Figure 5-1 Roller Bridge and Roller Nut

5.3 Tensioning System

There are many different ways to tension a string using a motor. Three specific approaches were considered in this case. One option involves attaching the string directly to the shaft of a worm gear arrangement, as shown in Figure 5-2. This method is simple and does not involve many moving parts. The motor is connected to a shaft with a screw (worm) which, when turned, forces the connecting gear to turn, causing the guitar string to wind around the gear shaft. As the gear rotates, the string's tension is altered according to the direction of rotation, leading to a change in frequency (pitch). These systems already exist on conventional guitar heads as the standard tuning system.

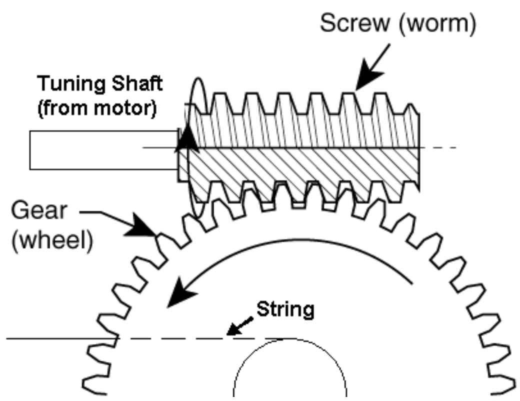


Figure 5-2 Worm Drive System

These systems are relatively simple, common and cheap. These devices also, by their design, cannot be driven from the reverse direction, that is to say that an input force applied to the gear wheel will not result in rotation of the screw. Consequently, string tension is maintained when no input force is present. The difficulty with this system is the complexity required to attatch the string to the shaft.

An alternative method requires a device that allows movement along the axis of the string in order to adjust string tension. This motion could be caused by the use of a conventional motor coupled with a threaded rod. A pictorial representation of this system is shown in Figure 5-3. The mechanical advantage inherent in such a screw system dramatically reduces the torque required to adjust string tension. Additionally, the screw would allow tension to remain on the string when power is disconnected. Furthermore, this system allows the easy connection and disconnection of guitar strings.

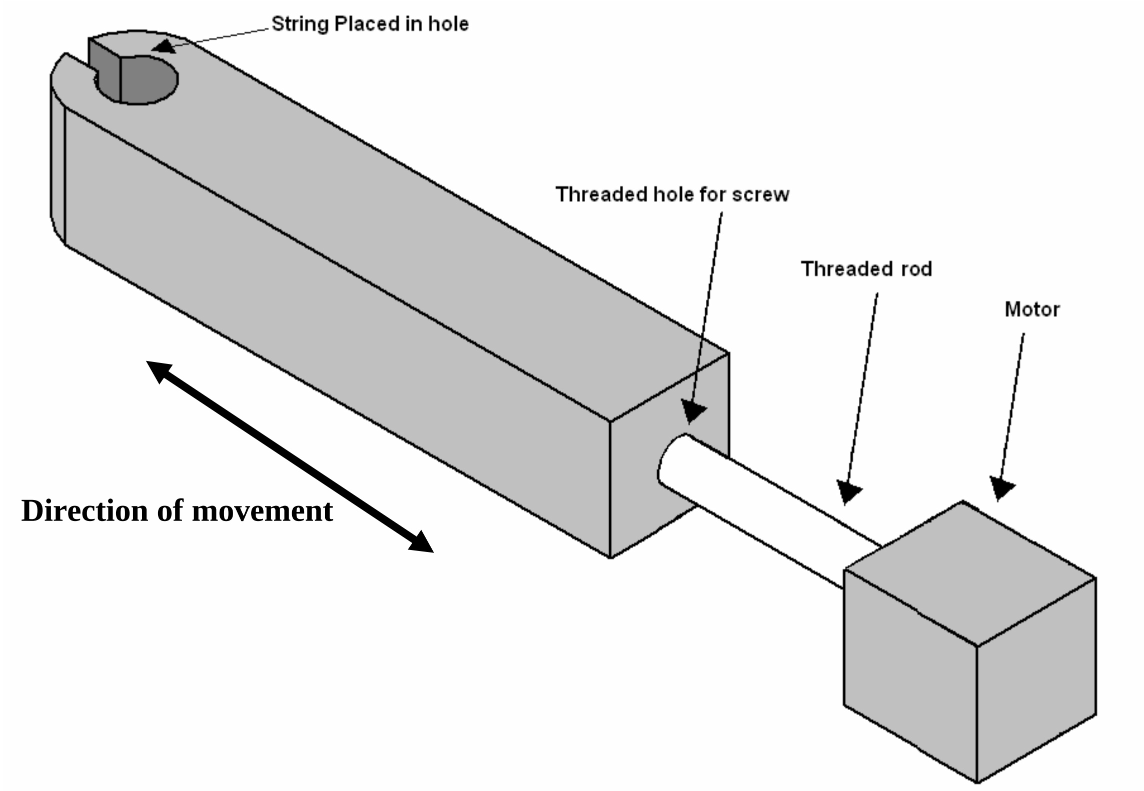


Figure 5-3 Alternative Tensioning system

A further alternative relies on a fulcrum-pivot system to tension the strings. This system utilizes a lever to gain mechanical advantage in order to reduce the torque required from the motor. This is achieved by attatching one end of the lever to the string, leaving the other end to be forced by a threaded block. This block travels along a threaded shaft, which is turned by a motor. As this motor turns, the position of the block changes, thereby controlling the position of the lever, and consequently the string tension. This design also allows tension to remain on the string whilst power is off is removed. With this system, connection and disconnection can be done with ease. A diagram of how this system works has been included and shown in Figure 5-4.

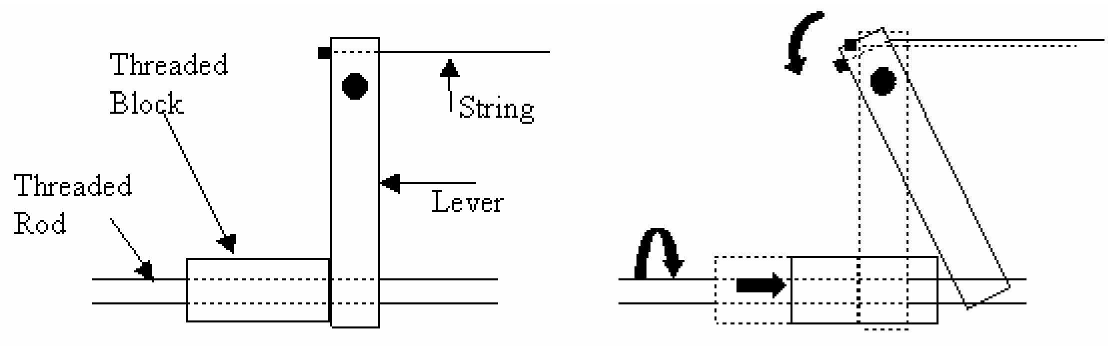


Figure 5-4 Diagram of lever arm system

The ability to withstand the high forces when holding the tension of the string is a requirement of any tensioning system, thus a suitably strong material must be selected to avoid failure of any components. To determine whether a component will fail under its normal load, finite element analysis can be applied, and the position and magnitude of maximum stresses identified. From the results of this analysis, an appropriate material can be selected. The most desirable material would be economic and lightweight.

5.4 Tensioning motors

To reduce overshoot, tensioning motors must be capable of spinning at low speeds. Inability to spin at low speeds could lead to not only overshoot, but possible string breakage. To avoid this a reduction gearbox is necessary. Additionally, the motors (or complete motor system, with gear box) must provide sufficient torque to tension the strings. In this application (string tensioning) the motors will have to be installed into the guitar. In order to actuate each string independently, six motors are required to be housed within the body of the guitar. Consequently, appropriately small motors must be selected.

Servo actuators are able to provide large amounts of torque, however the limitation of a finite travel distance (and hence a finite amount of energy that can be added to the system) renders them unsuitable in this instance. A more favorable option involves small electric motors with attatched gear-boxes, which is both economical and compact.

Of the actuators considered, the electric motor proves to be the most favorable option, as they are economical, reliable and readily available. The speed of an electric motor can be easily controlled by applying a varying voltage or a pulse train of varying duty cycle.

6 Data acquisition

6.1 Pickups

On electric guitars, string vibration is converted to an electrical signal by means of one or two coils. An iron ‘slug’ is placed under each string and a large magnet mounted behind the row of slugs. Very fine wire is wrapped around the set of slugs, several thousand times. When the strings vibrate, the magnetic field through the slugs is changed, inducing an AC signal in the coil. The whole assembly is called a ‘pickup’. A picture of a single coil pickup is shown in Figure 6-1.

Figure 6-1 Conventional single coil pickup

It is required for the chosen design, that a separate signal be obtained for each string. If each string were vibrating only at its fundamental frequency, signal separation could be realized with the use of simple bandpass filters. However, each string’s vibration contains many harmonics when plucked or strummed. Some of these harmonics exist at frequencies close to the fundamental frequencies of other strings. For example, the 4th harmonic of the low E string is in the same frequency range as the fundamental frequency of the high E string, therefore these two signals will be indistinguishable using only bandpass filters. For this reason it is necessary for the chosen design to incorporate a separate pickup, in conjunction with a bandpass filter for each string.

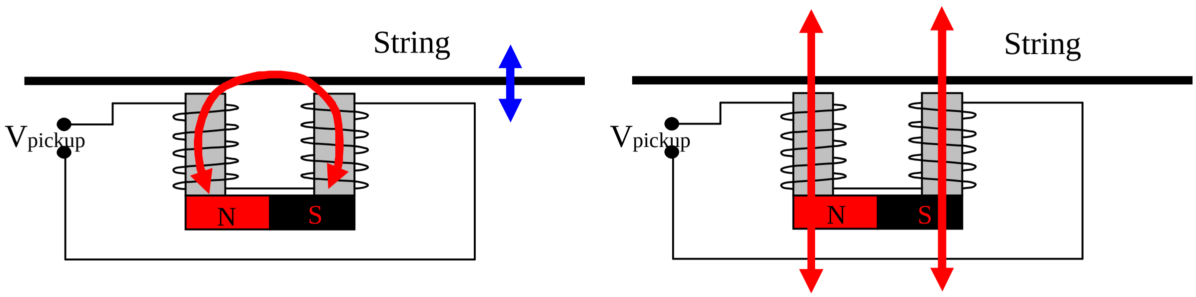


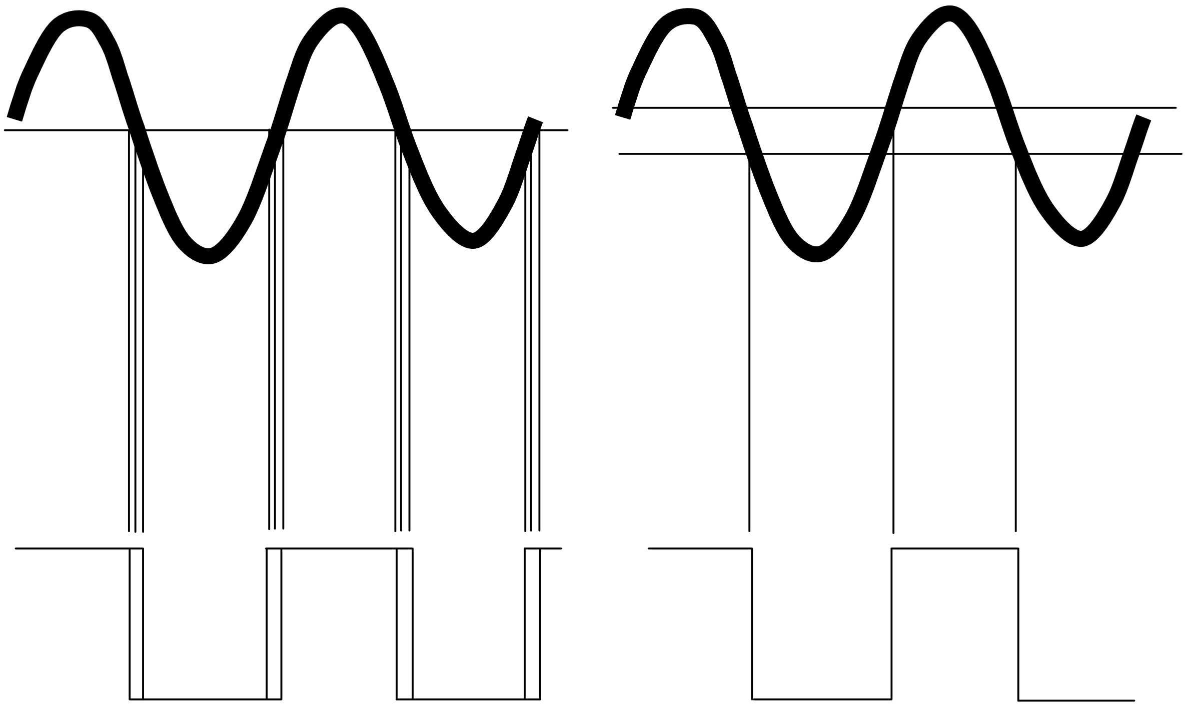
Figure 6-2 Humbucking Pickup Design

To reduce noise, a “humbucking” pickup design can be employed. A diagram demonstrating the function of this system is given above in Figure 6-2. As seen, two coils are wound in opposing directions. When the string vibrates, the change in flux in the magnetic circuit involving the string and the two coils induces a voltage across the two coils. Magnetic flux changes due to the environment (fluorescent lights, power supplies etc.), that pass through both coils in the same direction will induce opposing voltages in the two coils; hence the overall voltage will be zero. The overall voltage induced in both coils will therefore be relatively unaffected by environmental noise.

6.2 Bandpass filters

Assuming the fundamental frequency of the a string is close to a desired value, it can be isolated using a bandpass filter centered around this desired frequency. If each signal is sent through a bandpass filter, unwanted harmonics and signals from adjacent strings will be attenuated, while the fundamental frequency of the string will pass through unchanged. This reduces the frequency range over which the tuning system will be useful, however, this will not reduce its effectiveness as a fine tuning device as stated in the performance specifications.

The output signal of the bandpass filter is a sinusoid, which can be easily converted to a square wave using a Schmitt Trigger or comparator. This square wave could be sent directly to a digital input on a microcontroller. The period of the signal can then be determined by a microcontroller without the need for additional analogue to digital conversion or Fast Fourier algorithms.



a) b)

Figure 6-3 Noise rejection properties of a) Comparator b) Schmitt Trigger

As can be seen in Figure 6-3, any noise in the signal will result in extra zero crossings. If this signal is passed through a comparator, the zero crossings are still present in the output, whereas the thresholds inherent in a Schmitt Trigger remove any extra zero crossings. The Schmitt Trigger does, however, reduce the dynamic range of usable signal, as any signal with a peak value less than the threshold will not result in a usable output. Therefore, in order for a Schmitt Trigger to be beneficial, the thresholds chosen must be greater than the level of noise (unwanted harmonics or interference noise from other circuitry).

Schematics and a further explanation of the electronics can be found in appendix D

6.2.1 String Exciting device

If the guitar had the added functionality of causing the string to vibrate without external input, it would be completely automatic, which would go further in achieving the goal of requiring minimal user input. This requires a simple feedback loop involving the signal (originating at the pickup) being amplified and fed to a coil in close proximity to the guitar strings. If the transfer function of the feedback loop is greater than unity at a phase shift of 180°, the string (initially vibrating by background noise in the feedback loop) will vibrate with increasing magnitude, until a point of non-linearity (in either string or feedback loop) is reached. If the feedback loop includes the bandpass filter, the vibration of the string will be dominated by the fundamental frequency of the string.

Further research is required to identify the most efficient coil properties (wire gauge, number of turns etc.).

7 Control System

A microcontroller is used to create the closed loop control system. The microcontroller would be required to receive the output signal of the Schmitt Triggers. This then calculates the control action required (e.g. by way of a PID algorithm or similar), which in turn is used to drive the tensioning motors. Three options were identified:

- using one microcontroller to receive all 6 inputs and control all 6 motors
- using one microcontroller to receive inputs and a separate microcontroller to control the motors
- dedicating one microcontroller to each string (receiving 1 input and controlling 1 motor)

The first two options require the use of a real-time polling program to monitor all 6 inputs, thus decreasing the accuracy of the results, as the inputs cannot be monitored simultaneously. Therefore it was decided to investigate the third option, as it required less processor speed and fewer inputs and outputs per microcontroller.

7.1 Pitch Detection

The output from the analogue filtering circuitry will be a near square wave of a frequency equal to that of the string vibration. The bandpass filter will ensure that any harmonics of the fundamental frequency will be highly attenuated. It cannot, however, be assumed that the resulting signal will be free from harmonics and vibrations from adjacent strings. This dictates the need for methods of identifying and discarding unwanted results. The code responsible for pitch detection will be built on the assumption that signals indicating a frequency within 3 semitones of the desired pitch will be regarded as correct. All other signals will be considered invalid and will not be used to calculate the control outputs. For this reason when an invalid signal is received, the motors speed will be set to zero.

8 Finalised Design

8.1 Mechanical System

The design chosen for the mechanical system is a lever-linkage system. This choice was guided by many merits, including:

- Ease of manufacture
- Ease of assembly
- Geometric feasibility
- High torque capacity

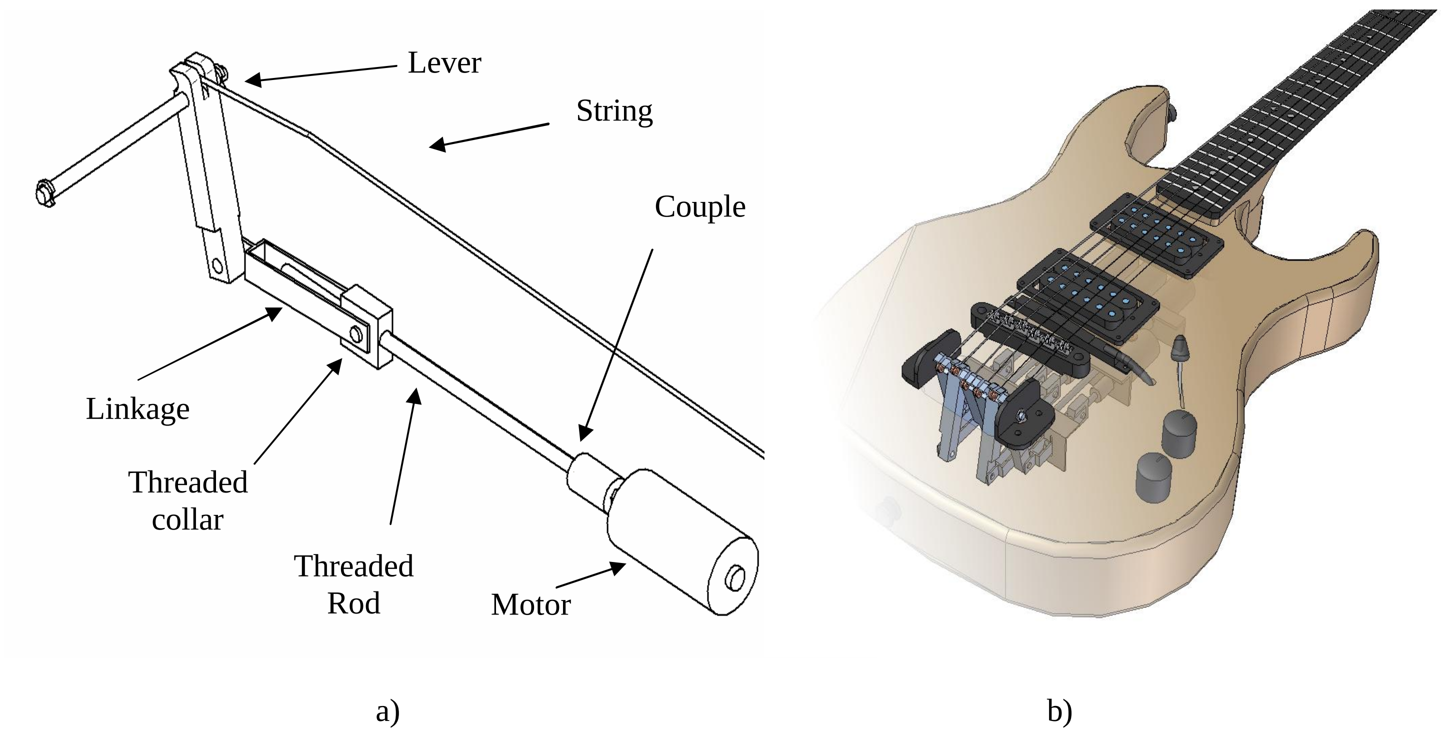


Figure 8-1 a) One string Model; b) System in Guitar

A schematic of a one string model is shown in Figure 4-1. The string tension is determined by the position of the lever. This lever is moved by a linkage, attached to a threaded collar. This collar is constrained to move along a threaded rod, which is coupled to the output of the motor/gear-box system. Therefore, the lever can be moved in either direction at different speeds by applying a varying voltage to the motor. This serves to increase or decrease the string tension accordingly. With the incorporation of a lever, linkage and threaded collar, minimal torque is required to adjust the tension of the string. A full analysis of torque produced by this system is given in appendix A.

8.1.1 Motor selection

The motor/gear-box arrangement is required to function at low speed with high torque, and have small dimensions and a voltage rating close to that of the rest of the electronics. A picture of the chosen motor is given in Figure 8-2.

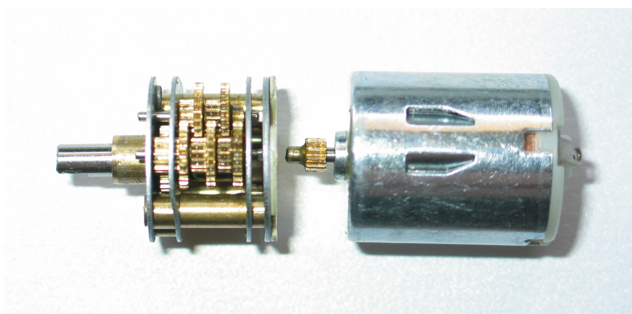


Figure 8-2 Motor with Gearbox

The motors are small enough to fit within the body of the guitar, without drastically altering the structural properties. The gear-box consists of a six stage reduction gear set, with each stage having a ratio of 12:25. This gives an overall reduction of approximately 82:1, with a maximum speed of 70 RPM and torque of 2.1kg·cm (0.2Nm).

In order to accommodate the motors within the body of the guitar, it is necessary for them to be interleaved, as shown in Figure 8-3.

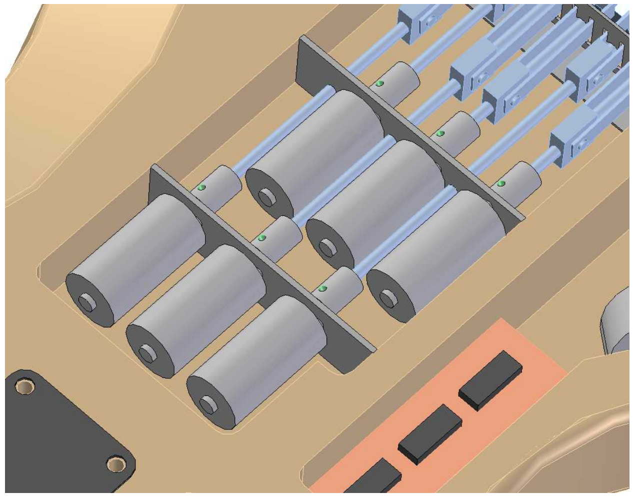


Figure 8-3 Interleaved Motor Arrangement

8.1.2 Stress analysis and material selection

Stress analysis was required to determine if the finalised model could withstand the high forces resulting from string tension, without yielding or deforming. Correct material selection was crucial in the complete design of the system, to ensure that all components remained durable under the normal loadings, without the need to replace parts periodically.

The component required to withstand the highest force, is the pivot shaft, which is the shaft around which all the tensioning levers rotate. Therefore, the force applied on this shaft is the total sum of all the forces from the strings, in addition to all the reacting forces from the linkages. To approximate the stress in the shaft resulting from these forces, a finite element analysis (FEA) program was employed (ANSYS).

The geometry, boundary conditions, and forces applied to the pivot shaft were modeled in ANSYS. The results from this modeling indicated that the maximum predicted stress in the shaft, at full loading, amounts to 605 MPa. Therefore, the most suitable material for this application is Steel Alloy 4340, provided it is heat treated to increase it’s strength. After proper heat treatment, the yield stress of this steel is approximately 870 MPa. With this yield stress, the pivot shaft has a safety factor of 1.4. This steel was chosen due to its low cost. The list of all suitable materials in this selection process is given in appendix B. The FEA analysis is shown in **Figure 8-4**. From this figure, it can be seen that the maximum stress is 605 MPa.

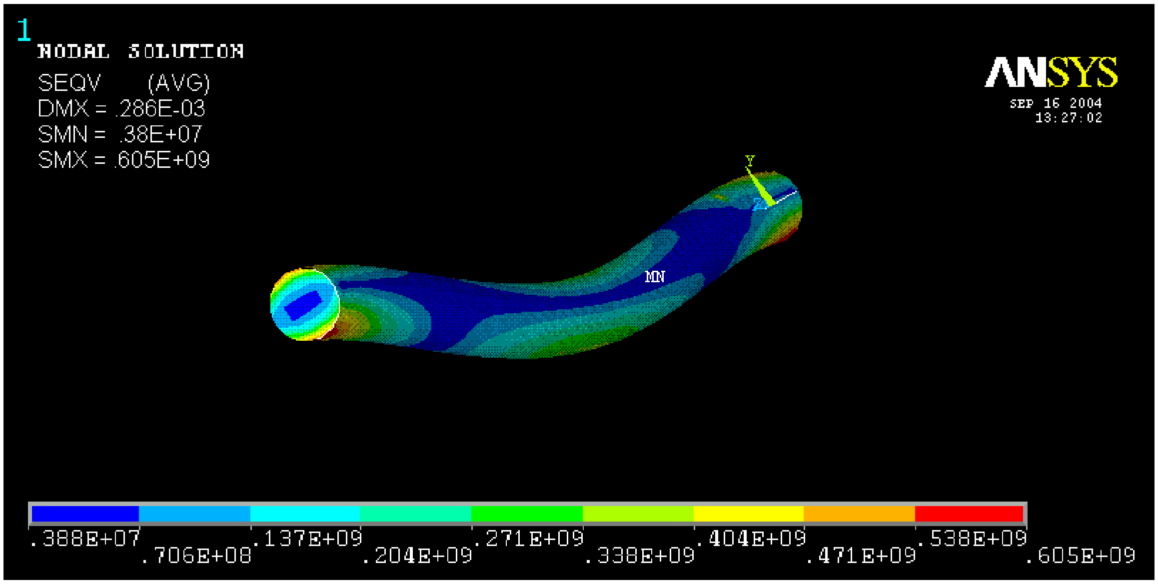


Figure 8-4 FEA simulation of pivot shaft

8.2 Data Acquisition

The data acquisition system, as previously mentioned, receives the vibration signal from a pickup mounted on the guitar body. A separate coil in the pickup is required for each string. This can be achieved through the use of a 'midi pickup'. A midi pickup provides a separate signal for each string, and interference from adjacent strings is almost non-existent. A picture of a midi pickup is given in Figure 8-5.

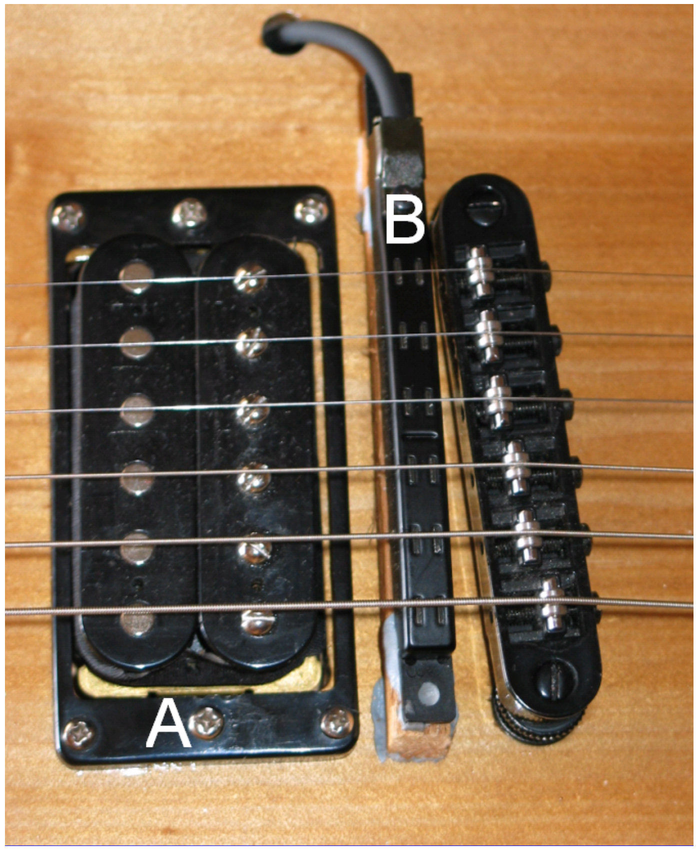


Figure 8-5 Conventional humbucking pickup (A), Midi Pickup (B)

This signal will be fed through a bandpass filter followed by a Schmitt trigger to turn the sinusoid input a square wave. This is a simple design, not involving complicated electronics. Furthermore, it is cheap in comparison to more sophisticated systems involving digital signal processors (DSPs).

8.3 Control System

Simplicity and a cost effectiveness were the two main foci in the selection process for the finalised system. The PIC16F628 was chosen as the basis for the control system due to it's low price and ease of programming (simple programming hardware and software was already owned). In the chosen design, an individual microcontroller (PIC16F628) is dedicated to each string. Each microcontroller receives one input and controls one output signal to the motor of its specific string.

The clock signal for the PIC16F628 can be generated in a number of ways. The PIC16F628 has an internal RC oscillator that can run at either 37 kHz or 4MHz. External crystals may be used to generate clock signal up to 20MHz. The three considerations when choosing a clock source were accuracy, resolution and cost. A higher speed clock provides greater resolution for measurements of frequency, whereas a more accurate (or more stable) clock source enables more accurate frequency measurements. An external clock source was necessary, as the internal oscillator only has an accuracy of 1%, which is unsuitable for tuning accuracies of 5% of a semitone. As the cost of external crystals is relatively independent of frequency, a 20MHz crystal was chosen to provide the greatest resolution and processing speed.

The near square wave, received from the Schmitt trigger is routed to an input with an interrupt-on-change feature. Within an interrupt subroutine, the time between consecutive rising and falling edges of the input signal is measure using an internal 16 bit timer. All measured times representing frequencies that fall outside the prescribed 6-semitone range are considered erroneous. If such a signal is encountered, the control system waits until a valid signal is encountered and then continues to average the frequencies. To obtain a valid detected pitch, only averages of a number of consecutive non-erroneous times are considered. To ensure that all valid periods do not cause the timer to overflow, the timer is run at half the speed of the processor (10MHz).

As the data acquisition system directly connects to the control system, and considering that each string has its own controller and signal processor, a one-stringed prototype controller was created implementing the analogue filtering and digital control on the same circuit board. This aided in keeping the entire system compact and cost effective, as it reduced the amount of circuit board that required etching. A picture of the filter and controller as well as the in-circuit programmer is shown below in Figure 8-6.

A simplified flow diagram of the control algorithm is shown in appendix C, and the finalised microcontroller code is shown in appendix G.

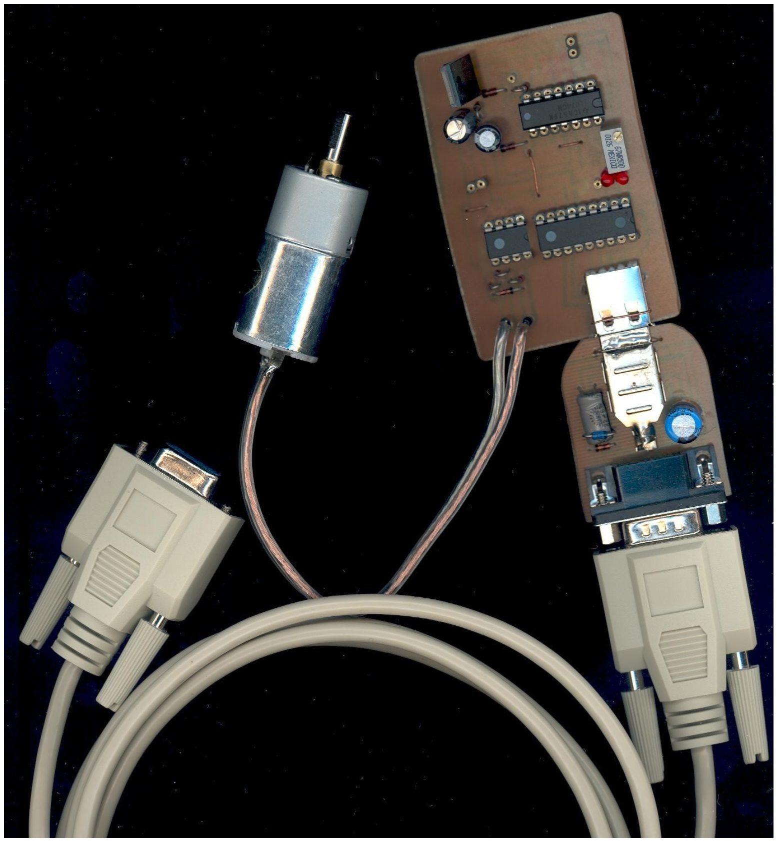


Figure 8-6 Controller circuitry

8.4 Prototype

For initial testing and tuning of the physical system, a prototype was built. The prototype, while simple, demonstrates that the task of a self-tuning instrument can be achieved. A one-string model was built (shown in Figure 8-7). This model included a means for manual tuning at one end of the string and a motor actuated tuning system at the other end. A pickup, bandpass filter, Schmitt Trigger and microcontroller system as described earlier was also built. This system allowed for initial practical tuning and testing of each component for suitability and effectiveness.

The one-string model was tested by attempting to tune a D string. This string was chosen for testing as it represented the greatest tension required. The system was able to tune the string to an accuracy of +/- 2% of a semitone, according to results obtained using a common electronic tuner. This accuracy is well within the desired accuracy and demonstrates the effectiveness of the control system.

The motor speed in the tensioning and slackening directions was almost identical, indicating that the string tension had little effect on the load of the motor. From this it can be seen that a motor with less torque would have been adequate. Although proportional control was implemented for very small tuning errors, the motor was predominantly run at full speed without any significant overshoot. Consequently, tuning speed was almost linearly dependant on the tuning error to be corrected. The tuning system was capable of tuning the D string from two semitones above (E) and from two semitones below (C) within 7 seconds. Tuning errors in the range of 10% of a semitone were generally able to be corrected in less than 2 seconds.

With this design, the most significant limitation is the tuning speed, which is dependant almost entirely on the motor speed.

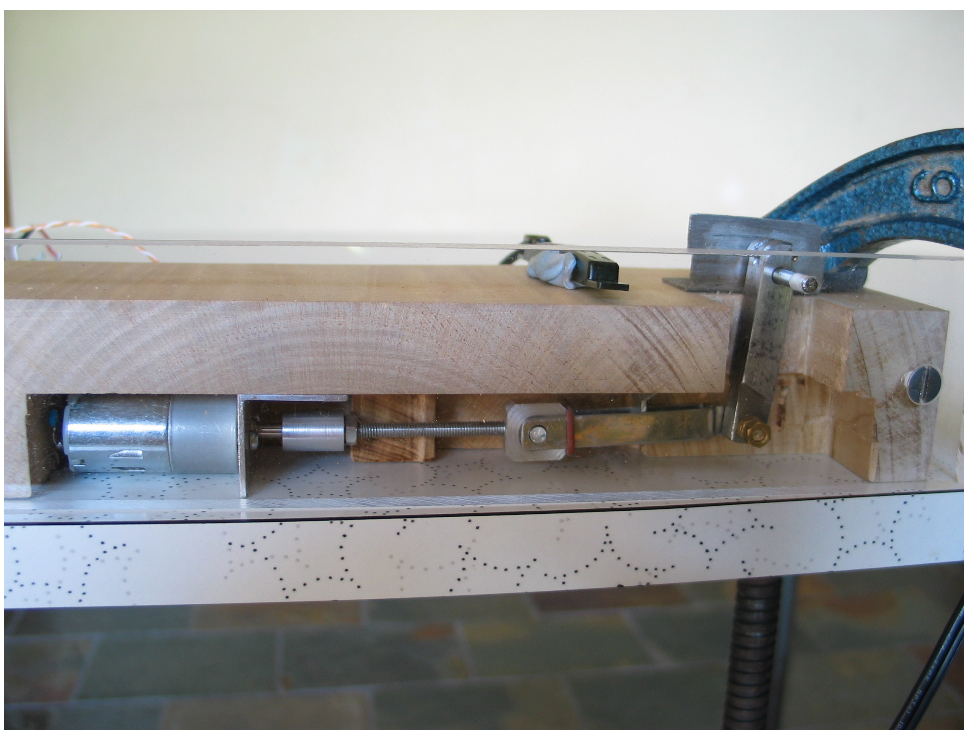


Figure 8-7 One String Model

8.5 Completed Model

A picture of the completed model is given in Figure 8-8.

Figure 8-8 Completed Guitar

8.5.1 Body and Neck Construction

To achieve the aim of building a guitar with comparable functionality and aesthetics, conventional guitar building techniques were used. The complete CAD model of the guitar was created before construction was commenced.

The guitar body was made from a single piece of Queensland Kauri Pine. The initial profile was cut using a band saw and then sanded. The edges of the profile were rounded using a router with a 5mm radius roller fitted bit. The cutouts for the neck seating, pickups and tuning system were achieved using a ruler guided router.

A standard Fender Stratocaster equivalent neck was purchased. The head was shaped using a jigsaw and finished with semi-gloss lacquer. The original plastic ‘nut’ was replaced with a roller nut. Off the shelf tuning pegs and string trees were installed on the head. The neck was attached to the body using a steel plate and four 2 inch wood screws.

A set of strings and a temporary tail piece were installed to help determine the correct position of the roller nut.

The body was sanded with 80, 120, 240 and 400 grit sandpaper and finished with four coats of high gloss Estapol.

8.5.2 Tuning Mechanism Construction

Keeping the overall weight of the tuning mechanism to a minimum was a major consideration in construction. The majority of parts were made from Aluminium, with the exception of the pivot shaft and the threaded rods.

8.5.3 Electronic Construction

The electronics were constructed on four separate boards (as shown in Figure 8-9) to allow for redesign of any components that did not achieve suitable functionality. The analogue filter circuitry was built on a double sided board. Due to the large number of components, the filters were implemented using 0803 surface mount parts. This dramatically reduced the board size required to fit all 6 filter sets. Special attention was paid to ensuring that the common ‘virtual earth’ tracks were wide and covered with ample solder. By reducing the resistance of such common tracks, interference between filter sets is reduced.

The microcontroller board, motor driver board and power supply were built using standard pitch vero-board to reduce construction time.

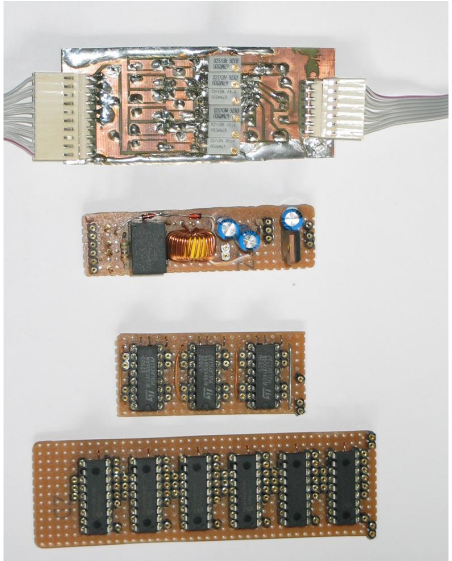


Figure 8-9 Analogue filters, Power supply, Motor drivers and Microcontrollers

8.5.4 Results

With no power applied to the tuning system, the guitar displayed functionality comparable to a conventional electric guitar. From the front, the guitar looks similar to many guitars on the market, with the midi pickup and lever arrangement being the only minor visual differences. With a weight of 4.5 kg, it is comparable to lighter electric guitars. This low weight can be attributed to the volume of wood removed to install the tuning system. When played, the tone and sustain of string vibration was not noticeably different from most guitars.

When power was first applied to the tuning system, before any strings were plucked, it was noticed that one of the motors was already turning. From examination of the outputs of the corresponding microcontroller, it was clear that it was receiving signal that were considered valid (within 3 semitones of the desired pitch). Examination of the input to the microcontroller revealed a relatively consistent square wave with a frequency of 200Hz. It was assumed that the source of this signal was the 4th harmonic of mains induced noise. Two solutions to this problem were identified; adjusting the Schmitt Trigger thresholds, or reducing the noise in the circuit. A regulated power source was used to eliminate mains induced noise and hence eliminate the erroneous signals.

Due to the difference in properties in each string, the relationship between string elongation and the corresponding tension change is not constant for all strings. For this reason, the time taken to correct similar tuning errors on different strings is not uniform. For example, the tuning system is able to adjust the pitch of the high E string by a semitone, in a time of ten seconds. A similar pitch adjustment of the low E string, however, can be achieved in less than six seconds.

When all strings are strummed, the microcontrollers receive valid signals for approximately five seconds. Therefore, if any strings have a tuning error of more than 50% of a semitone, the strings will need to be strummed more than once to give the tuning system enough time to correct the tuning.

9 Conclusions & Future Work

9.1 Conclusions

A guitar capable of tuning itself has been designed and built. The guitar built is capable of simultaneously tuning all 6 strings to an accuracy of at least 5% of a semitone. The time taken to achieve this accuracy depends on the amount of tuning error to be corrected. Given that a well maintained guitar will not be subject to tuning deviations of greater than 20% of a semitone over the period of an hour of normal playing, retuning using this system will take less than 10 seconds to perform. The only user input required by the system is that the strings be plucked or strummed.

The aims of this project were successfully completed. Such a system would greatly benefit guitar players who perform in situations where it is not feasible or favorable to manually tune their instrument during the performance. Such tuning is possible with this system with minimal concentration and only requires the use of one hand.

9.2 Future Work

There are many avenues for improving the current system. Improvements to the size, performance and autonomy will be discussed.

9.2.1 Decreased size

One of the major difficulties in marketing a self tuning mechanism is the amount of alteration required to the guitar body, in order to install the system. Decreasing the size of the electronic and mechanical systems would make the system easier to install and more marketable. The electronic system could be made more compact by combining all circuits onto one double sided PCB. The use of more surface mount components could further reduce PCB size.

It has been noted previously that the chosen motor/gear-box arrangement has much more torque than is required. By choosing smaller motors, the mechanical system could be made to take less space. Different geometric configurations could be investigated to optimize the amount of space required.

9.2.2 Increased Performance

Currently, the tuning speed is limited by the speed of the motors. If a different gear-box were employed, this speed could be increased and the overall performance improved.

9.2.3 Increased Autonomy

The current system has no means by which to ensure that the actuation limits are not reached. It is therefore possible to unknowingly cause damage to the system while trying to tune the guitar. Ideally, two sets of limit switches would be employed to determine if any of the levers were near their limit of travel.

Due to time limitations, a system to automatically vibrate the strings was not designed. Such a system would further decrease the effort required on the part of the user. Additionally, the feasibility of an automatic string vibrator was not looked into, in regards to cost and functionality.

Appendix A

The maximum tension the tensioning system has to endure is determined below.

There relationship of velocity of a transverse wave in a stretched string or cord is defined in the following equation:

$$v = \sqrt{\frac{F_T}{\mu}} \qquad \text{(eq 1)}$$

(Source: Eq. 15-2 Giancoli[2])

Where:

v is the velocity of the transverse wave (in m/s)

F_T is the force on the string (tension) (in Newtons)

μ = m/l, the mass per unit length of the string (in kg/m)

Velocity is a function of wavelength of a wave, and its frequency, outlined below:

$$v = \lambda f \qquad \text{(eq. 2)}$$

(Source: Eq 15-1 Giancoli[3])

Where:

λ is the wavelength of the transverse wave of a string (m)

f is the frequency of the vibrating string (Hz)

For a vibrating string, the transverse wave's wavelength is defined in the following equation:

$$\lambda_n = \frac{2L}{n} \quad n = 1,2,3 \ldots \qquad \text{(eq. 3)}$$

(Source: Eq 15-17 Giancoli[4])

Where:

L is the length of string (m)

n is the number of the harmonic

[2] Giancoli, Douglas C. (2000), *Physics for Scientists & Engineers*, Prentice-Hall: New Jersey, Page 392
[3] Giancoli, Douglas C. (2000), *Physics for Scientists & Engineers*, Prentice-Hall: New Jersey, Page 390
[4] Giancoli, Douglas C. (2000), *Physics for Scientists & Engineers*, Prentice-Hall: New Jersey, Page 406

In this calculation, the first harmonic is to be used (i.e. n=1), because this will be the wavelength at the fundamental frequency.

Substituting eq. 3 into eq. 2 gives:

$$v = 2Lf \qquad \text{(eq. 4)}$$

Substituting eq. 4 into eq. 1 and re-arranging gives

$$f = \frac{1}{2L}\sqrt{\frac{F_T}{m}} \qquad \text{(eq. 5)}$$

Equation 5 shows the relationship between frequency (Hz) and Force (tension, N)
Re-arranging to get the tension force as the subject gives:

$$F_T = 4L^2 f^2 m \qquad \text{(eq. 6)}$$

The force in each string can be found with this relationship.

The value of μ changes with different gauges of guitar string. The larger the gauge, the more tension will be on the string. To calculate the maximum force on each string, the highest gauge string must be used in calculations. Therefore, the properties of high gauge guitar strings must be obtained. These were found by measuring the weight of each high gauge string (one for each note), and then dividing it by the length of the string, Giving μ (mass per unit length) for that specific string. The length of the guitar string used in the following calculations is 65cm (0.65m).

The Properties for each high gauge string are the following:

Note	E	B	G	D	A	E
• (Hz)	330	247.22	196.22	146.82	110.12	82.5
Mass (kg)	3.44E-04	6.05E-04	1.17E-03	1.97E-03	3.71E-03	6.15E-03
• (kg/m)	5.29E-04	9.31E-04	1.80E-03	3.03E-03	5.71E-03	9.46E-03

Using the collected data, it is possible to calculate the tension in all of the strings. Therefore, the calculated tension in the strings using eq 5. is the following:

Note	E	B	G	D	A	E
Tension (N)	97.40	96.14	117.12	110.41	116.97	108.83

The Maximum tension is 117 N.

The maximum torque produced by system is shown below.

Deriving the ratios of displacement:

For the collar and threaded rod:

1 revolution = 0.7 mm of travel.

= 4• mm (rotation) = 0.7 mm of travel

therefore, reduction of distance = $\frac{4p}{0.7}$ = 18:1

As torque coming out of motor/gear-box = 2.1 kg.cm

From threaded rod-collar relationship (reduction 18:1). torque output from collar (assuming no friction losses),

= 18x2.1 = 37.8 kg

leverage gained from the lever (5:1 ratio) = 5x37.8

= 189kg • 1854 N.

Max String Tension = 117N < 1854N max theoretical force available.

Appendix B

Here is the database for all materials with a yield strength above 607MPa.

Name	Material Condition	•	E	• y	TS	$/kg
Steel alloy 4140	Bar; normalized @ 870°C	7.85	207	655	1020	1.75-1.95
Steel alloy 4140	Quenched/tempered @ 315°C	7.85	207	1570	1720	
Steel alloy 4340	Bar; normalized @ 870°C	7.85	207	862	1280	3.3
Steel alloy 4340	Quenched/tempered @ 315°C	7.85	207	1620	1760	
Stainless alloy 440A	Quenched/tempered @ 315°C	7.8	200	1650	1790	
Ductile Iron; grade 120-90-02	Quenched/tempered; high prod.	7.1	164	621	827	1.45-1.85
Ductile Iron; grade 120-90-02	Quenched/tempered; low prod.	7.1	164	621	827	3.30-5.00
Titanium alloy Ti-5A1-2.5Sn	Annealed (equiaxed grains)	4.43	114	830	900	55.00-130.00
Titanium alloy Ti-6A1-4V	Soln. heat treated/aged	4.43	114	1103	1172	

Legend	
•	Density of material (kg/m^3)
E	Youngs Modulus (Gpa)
• y	Yield Stress (MPa)
TS	Tensile Strength (Mpa)
$/kg	Price of material per kg.

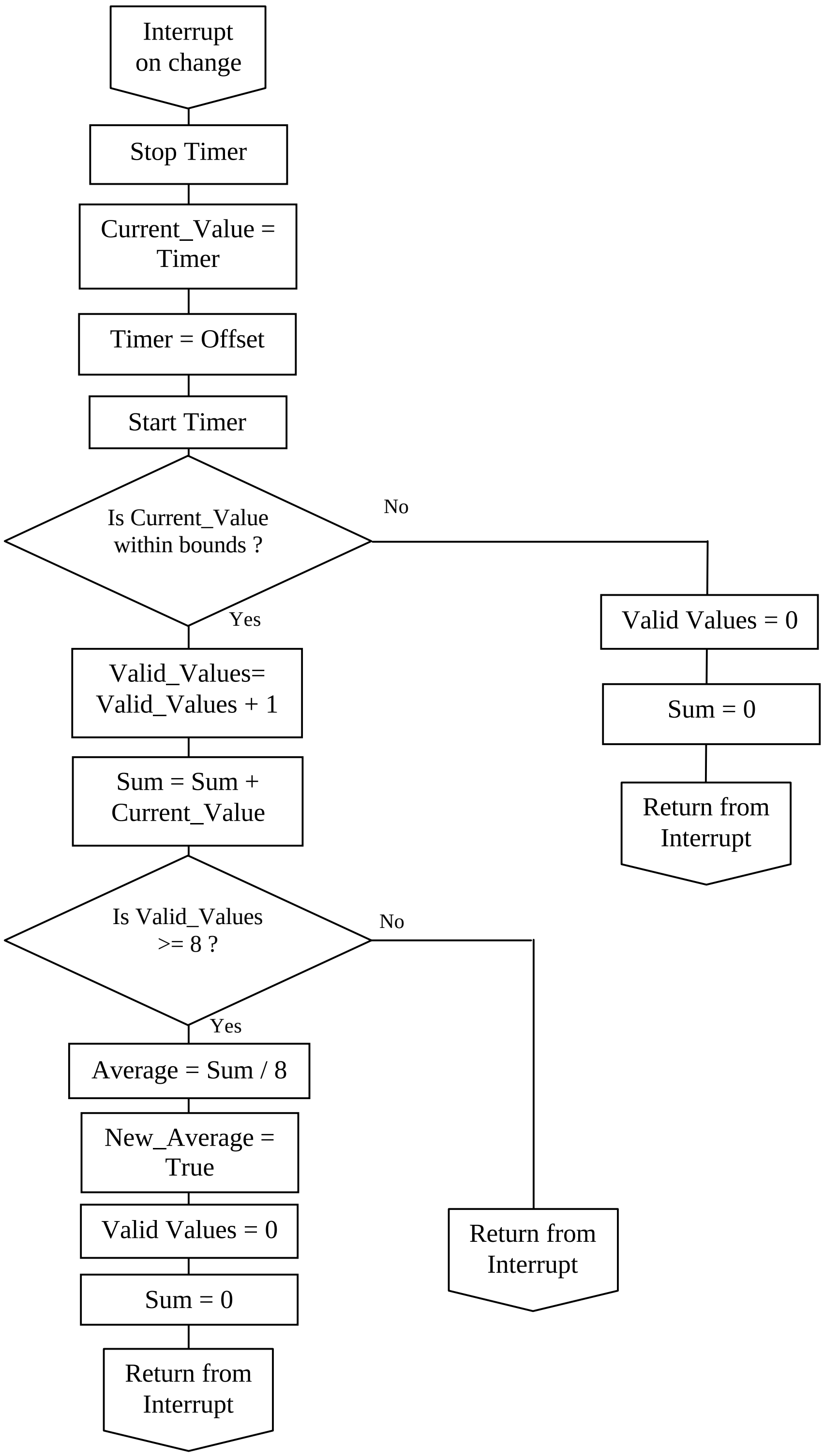
This Data was obtained from the material selection database from the CDROM of the following text book.

Callister, William C. (2001), *Fundamentals of materials science and engineering : an interactive e.text*, John Wiley: New York

Appendix C

Figure 1 shows a simplified flow diagram of the interrupt subroutine called when the digital value at the control input to the microcontroller changes.

The 16 bit timer should be stopped before it is read. The timer comprises of two 8bit values which must be read individually. If the timer is not stopped before reading, the lower byte may roll over between reading the lower and upper bytes, causing an erroneous value. After the timer has been read, it is reset to an offset value which compensates for the time while the timer was off before restarting. Each value is checked to see if it lies within the valid bounds. When 8 valid values are accumulated, they are averaged and a flag is set to alert the main program that a new valid average is available. The complete control code can be found in appendix G.



Interrupt on change
Stop Timer
Current_Value = Timer
Timer = Offset
Start Timer
Is Current_Value within bounds ?
No
Yes
Valid Values = 0
Sum = 0
Return from Interrupt
Valid_Values= Valid_Values + 1
Sum = Sum + Current_Value
Is Valid_Values >= 8 ?
No
Yes
Average = Sum / 8
New_Average = True
Valid Values = 0
Sum = 0
Return from Interrupt
Return from Interrupt

Figure 1

Appendix D

The output from each pickup coil (L_1) is amplified to a usable level, by a simple inverting amplifier. The gain of this system is set by the ratio of R_2 to R_1. R_2 may be replaced by a variable resistor for easy calibration. R_1 is chosen to match the impedance of the pickup (50•). The amplified signal is then sent to a second order bandpass filter. The gain of the system is given by:

$$A_{f_0} = \frac{R_5}{2R_3}$$

The centre frequency is given by:

$$f_0 = \frac{1}{2\mathrm{p}C_1\sqrt{\frac{R_3R_4R_5}{R_3+R_4}}} \quad \text{where } C_1 = C_2$$

The Quality factor (ratio of bandwidth to centre frequency) is given by:

$$Q = f_0C_1R_5$$

It is desirable that the bandpass filter have a high Q-factor, so as to highly attenuate harmonics and vibrations from adjacent strings.

The component values are chosen, using Excel, by setting desired gain, frequency and Q-factor, and varying the dependant variable C_1, until all component values lie within realistic available values. (see Appendix E). The filter can be tuned by replacing R_4 with a multiple-turn variable resistor.

The filtered signal is amplified to saturation by a Schmitt trigger (figure 3). The positive feedback (R_7), will ensure that the output of the op-amp remains at the supply voltage (either 0V or 9V), until the input crosses a threshold determined by R_6 and R_7. The input threshold is the point at which the positive terminal of the op-amp changes sign with respect to the negative terminal (connected to ‘earth’). The voltage at the positive terminal is given by:

$$\Delta V_{+} = V_{output} - \frac{V_{ouput} - V_{input} * R_7}{R_6 + R_7}$$

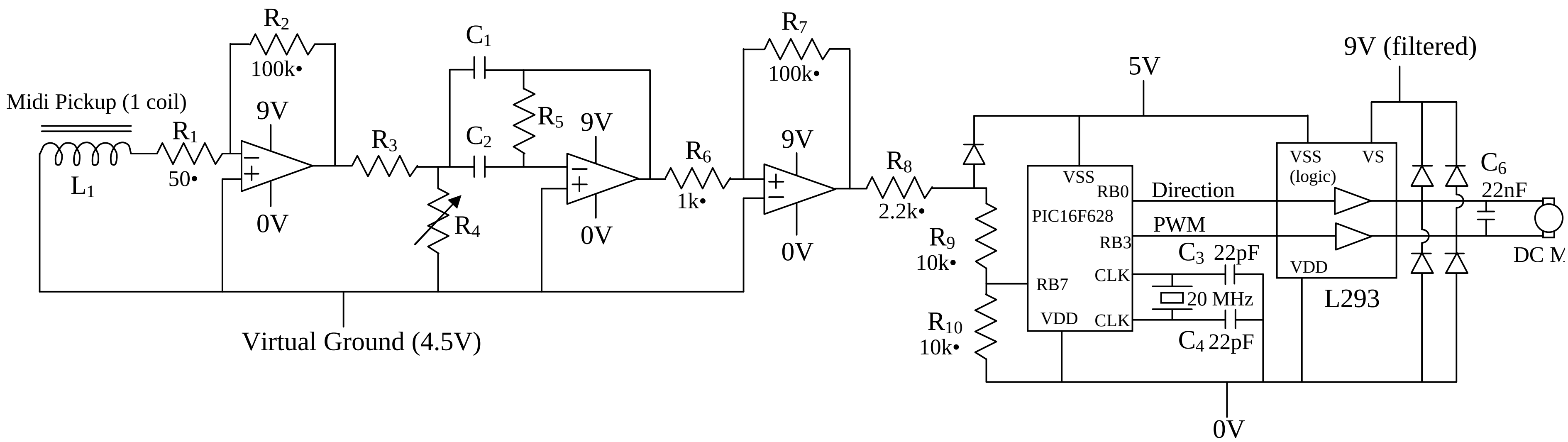
Thus, for a 9V supply (4.5V virtual earth) and R_6=1k• , R_7=100k• , the thresholds would be +/- 45mV of ground. The diode is used to reduce the maximum voltage of this square wave to 5V (supply voltage) +0.7V (diode voltage). The minimum voltage is determined by the lower limit of the TL074 output (around 1V). The input to the PIC16F628 must be below 0.8V to register a logic low input and above 2V to register a logic high voltage. Using the voltage divider network (R_9 and R_{10}), the 5.7V to 1V square wave is reduced to a 2.35V to 0.5V square wave. This provides noise rejection up to 350mV.

The PIC16F628 derives its clock signal from a 20MHz crystal. This crystal requires stabilization in the form of two 22pF capacitors connecting the oscillator pins to ground.

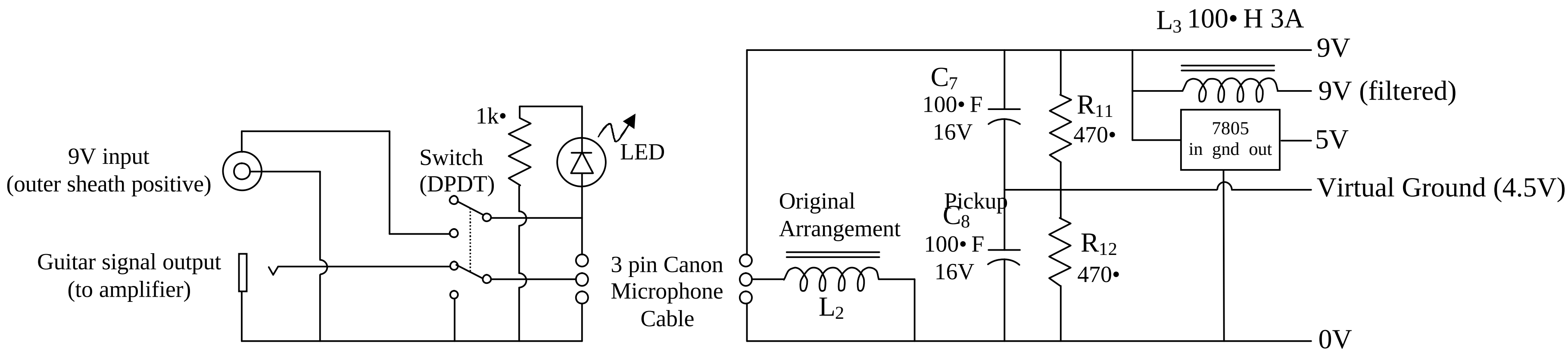
The two motor outputs from the PIC16F628 (controlling direction and speed) are fed into the inputs of a push-pull driver chip (L293). For driving inductive loads, the driver chip requires diodes clamping all outputs to within the supply rails. These diodes do not dissipate much energy and can therefore be silicon signal diodes. A 22nF capacitor is also placed in parallel with the motor to reduce voltage spikes cause by the inductive load. The third measure to reduce interference cause by the motors is a 100• H inductor placed in series with the positive motor power supply.

All the audio circuits described, require a bi-polar power supply. This can be accomplished by creating a virtual ground, where R_9 = R_{10}. These resistor values should be chosen to be at the minimum required to keep ground stability. The supply is chosen as 9V to be consistent with the vast majority of guitar-related electronic systems.

Power is supplied to the guitar via a 3 pin Canon microphone socket. Power is conveyed from a 9V mains adaptor through a foot operated switch and a microphone cable to the guitar. The foot operated switch (the most convenient switching mechanism when both hands are on a guitar) is a heavy duty DPDT switch which also acts to mute the guitar output while the tuning mechanism is operating. The operation of the tuning mechanism is indicated by an LED situated in the foot switch housing.



- This circuit represents the control system for one string. Each of the 6 control systems have different values for C_1, C_2, R_3, R_4 and R_5
- All power outputs in the circuit below are common to each of the control systems



- All diodes are 1N4149 signal diodes
- All op-amps are part of quad op-amps (TL074)
- The L293 driver chip is a 4 channel push pull driver, thus each package is used for 2 strings

Appendix E

Bandpass Filter Component Selection

Given Filter Parameters				Calculated Resistor Values			Available Component Values				Resulting Filter Parameters		
Frequency	Gain	Q-factor	C	R5	R3	R4	R5	R3	R4	C	Frequency	Gain	Q-factor
82.5	10	20	0.68	113,479	5,674	72	120,000	5,600	68	0.68	82.49028	11	21
110.12	10	20	0.47	123,003	6,150	78	120,000	6,800	80	0.47	110.0687	9	20
147	10	20	0.47	92,144	4,607	58	100,000	4,700	54	0.47	146.961	11	22
196.22	10	20	0.22	147,474	7,374	93	150,000	6,800	92	0.22	196.1594	11	20
247.22	10	20	0.22	117,051	5,853	74	120,000	5,600	72	0.22	247.1857	11	21
330	10	20	0.1	192,915	9,646	122	220,000	10,000	107	0.1	330.0882	11	23

Appendix F

Microprocessor crystal configuration selection

			Internal RC Oscillator			External Crystal			External Crystal		
			4,000,000 Hz	Oscillator frequency		**10,000,000** Hz	Crystal frequency		**20,000,000** Hz	Crystal frequency	
			1,000,000 Hz	Instruction frequency		2,500,000 Hz	Instruction frequency		5,000,000 Hz	Instruction frequency	
			1 • s	Instruction period		0.4 • s	Instruction period		0.2 • s	Instruction period	
			1 % oscillator error ***			0.003 % crystal error			0.003 % crystal error		
			3 instruction error **			3 instruction error			3 instruction error		
Note	Freq	Period	Instructions *****			Instructions			Instructions		
-3 semitones	69.4	0.0144	14415	0.5%	17.5%	36037	0.2%	0.07%	72075 +	0.1%	0.07%
E	82.5	0.0121	12122	0.5%	17.5%	30304	0.2%	0.10%	60608	0.1%	0.07%
+3 semitones	98.1	0.0102	10193	0.5%	17.4%	25483	0.2%	0.09%	50965	0.1%	0.06%
-3 semitones	92.6	0.0108	10799	0.5%	17.4%	26998	0.2%	0.10%	53996	0.1%	0.07%
A	110.1	0.0091	9081	0.8%	17.6%	22702	0.3%	0.09%	45405	0.2%	0.09%
+3 semitones	131.0	0.0076	7636	0.7%	17.5%	19091	0.3%	0.10%	38181	0.1%	0.06%
-3 semitones	123.6	0.0081	8090	0.7%	17.5%	20226	0.3%	0.11%	40451	0.1%	0.07%
D	147.0	0.0068	6803	0.8%	17.4%	17008	0.4%	0.14%	34016	0.2%	0.09%
+3 semitones	174.8	0.0057	5721	1.1%	17.6%	14302	0.5%	0.17%	28604	0.2%	0.10%
-3 semitones	165.0	0.0061	6061	1.1%	17.7%	15152	0.4%	0.09%	30305	0.2%	0.09%
G	196.2	0.0051	5097	1.2%	17.6%	12742	0.5%	0.14%	25483	0.2%	0.07%
+3 semitones	233.3	0.0043	4286	1.5%	17.7%	10714	0.6%	0.13%	21429	0.3%	0.13%
-3 semitones	207.9	0.0048	4811	1.3%	17.6%	12027	0.5%	0.13%	24053	0.2%	0.06%
B	247.2	0.0040	4045	1.4%	17.5%	10113	0.5%	0.07%	20226	0.3%	0.07%
+3 semitones	294.0	0.0034	3402	1.9%	17.7%	8504	0.6%	0.08%	17008	0.3%	0.08%
-3 semitones	277.5	0.0036	3604	1.9%	17.8%	9010	0.7%	0.21%	18020	0.3%	0.11%
E	330.0	0.0030	3031	2.0%	17.7%	7576	0.8%	0.14%	15153	0.4%	0.14%
+3 semitones	392.4	0.0025	2548	2.3%	17.7%	6371	1.1%	0.32%	12742	0.5%	0.19%

* - This represents the max number of instruction cycles which can be executed between consecutive rising edges of the filtered guitar signal
** - This represents the error introduced by possible delays entering and exiting interrupt subroutines
*** - All errors in calculated frequency are expressed in percentage of a semitone
\+ - This value is outside the bounds of the 16-bit timer (max value = 65535)

Appendix G

This is the control code responsible for the high E string. Each control system is identical, with the exception of the declared constants with correspond to the lower bound, range and half-range of valid periods for each given string.

```
LIST p=16F628,r=DEC     ; Put assembler into PIC16F628 mode.
                        ; r=DEC means decimal numbers are
                        ; assumed if 'B' or 'h' not specified.

      #include <P16F628.INC>
;**************Declare Constants************************

LB_H              equ   12    ; Lower bound of valid period
LB_L              equ   113   ; low bit
Range_H           equ   5     ; Range of valid periods
Range_L           equ   40    ; low bit
hr_H              equ   2     ; Half-Range of valid periods
hr_L              equ   205   ; low bit

;**************Declare Variables************************

Temp_W            equ   32
Temp_STATUS       equ   33
subSTATUS         equ   34
Valid_Count       equ   35
Valid_flag        equ   36
Sum_H             equ   37
Sum_L             equ   38
PA                equ   39
PB                equ   40
currentspeed      equ   41
desiredspeed      equ   42
EXOR              equ   43
lastportb         equ   44

;**************Initialise interrupt subroutine**********

      goto InItSeTuP
      ORG 4

INTERRUPT
      call  clearRBIF         ;Clear the flag causing the interrupt
      btfsc PIR1,TMR1IF       ;Did timer overflow?
      goto  timeout           ;GoTo: timeout
      movf  PORTB,W           ;
      iorwf lastportb,F       ;has the signal changed or was it just
      btfss lastportb,7       ;a spike caused by motor interference
      retfie                  ;Return From Interrupt
      movf  PORTB,W
      movwf lastportb         ;Assign:lastportb = PORTB
      btfsc PORTB,7           ;Rising or falling edge?
      retfie                  ;Return From Interrupt
      bcf   T1CON,TMR1ON      ;Stop Clock
      movwf Temp_W            ;Save W register
      movf  STATUS,W
      movwf Temp_STATUS       ;Save STATUS register
      movf  TMR1H,W
```

```
      movwf A_H                 ;read timer
      movf  TMR1L,W
      movwf A_L                 ;read timer
      clrf  TMR1H               ;Offset Reset
      movlw 11
      movwf TMR1L               ;Offset Reset
      bsf   T1CON,TMR1ON        ;Start Clock
      bcf   STATUS,C            ;Set: STATUS Bit C OFF
      rrf   A_H,F               ;divide by 2
      rrf   A_L,F               ;Calc: A_L =  >> A_L
      call  subLB               ;subtract Lower bounds from time
      btfsc subSTATUS,1         ;if -ve, then invalid
      goto  tooquick            ;GoTo: tooquick
      call  subRANGE            ;subtract range from time
      btfss subSTATUS,1         ;if not -ve, then invalid
      goto  toolong             ;GoTo: toolong
      bsf   PORTB,6             ;Set PORTB 6 ON to show valid signal
      bsf   Valid_flag,1        ;new valid value
      call  addHALFRANGE        ;add half range, so desired time=0
      call  Store_Value         ;store in allocated memory
      decfsz Valid_Count,F      ;have we got 8 values yet?
      goto  Exit_Routine        ;GoTo: Exit_Routine
      movlw 8
      movwf Valid_Count         ;reset pointer
      bsf   Valid_flag,0        ;we have 8 valid values
      goto  Exit_Routine        ;GoTo: Exit_Routine
timeout                         ;Label
      bcf   T1CON,TMR1ON        ;Stop Clock
      movlw 255
      movwf TMR1L               ;set stopped clock to full
      movlw 255
      movwf TMR1H               ;so next time will be too large
      bcf   PIR1,TMR1IF         ;clear the interupt
      goto  Invalid             ;GoTo: Invalid
toolong                         ;Label
      bcf   PORTB,1             ;Set: PORTB Bit 1 OFF
      goto  Invalid             ;GoTo: Invalid
tooquick                        ;Label
      bsf   PORTB,1             ;Set: PORTB Bit 1 ON
Invalid                         ;Label
      bcf   PORTB,6             ;Set PORTB 6 OFF to show invalid signal
      movlw 8
      movwf Valid_Count         ;reset pointer
      clrf  Valid_flag          ;no valid or new valid values
Exit_Routine                    ;Label
      movf  Temp_STATUS,W
      movwf STATUS              ;Restore STATUS
      movf  Temp_W,W            ;Restore W
      retfie                    ;Return From Interrupt

;*************Initialise Ports************************
InItSeTuP

      __CONFIG  B'11111100000010'

      CLRF PORTA
      CLRF PORTB
```

```
      BSF STATUS, RP0
      MOVLW B'10000000'
      MOVWF OPTION_REG
      MOVLW B'11111111'
      MOVWF TRISA
      MOVLW B'10000100'
      MOVWF TRISB
      BCF STATUS, RP0

;**************Start Of Main Program*******************

START             ;Start of Main Program
      movlw 7
      movwf CMCON              ;turn off comparators
      clrf  PORTA              ;clear port A
      clrf  PORTB              ;clear port B
      clrf  Valid_flag         ;clear flags
      clrf  currentspeed       ;Assign:currentspeed = 0
      call  setuptimer1        ;setup and start 16bit timer
      bsf   INTCON,RBIE        ;setup int-on-change
      call  setuppwm           ;Calls Subroutine: setuppwm
Motor_off                      ;Label
      clrf  desiredspeed       ;Assign:desiredspeed = 0
Main_Loop                      ;Label
      call  desirable          ;Calls Subroutine: desirable
      call  makemotorgonow     ;Calls Subroutine: makemotorgonow
      btfss Valid_flag,0       ;is there 8 valid values?
      goto  Motor_off          ;GoTo: Motor_off
      btfss Valid_flag,1       ;is there new valid values?
      goto  Main_Loop          ;GoTo: Main_Loop
      bcf   Valid_flag,1       ;no more new valid values
Calc_average                   ;Label
      call  Calc_Average       ;Calculate average deviation
Motor_stuff                    ;Label
      call  dispmotor          ;display tuning
      goto  Main_Loop          ;GoTo: Main_Loop
      goto  $                  ;End of Main Program

;**************Subroutines******************************

setuptimer1
      movlw B'11000000'
      movwf INTCON             ;setup interrupts
      movlw B'00010001'
      movwf T1CON              ;setup timer
      bsf   STATUS,RP0         ;Set: STATUS Bit RP0 ON
      bsf   PIE1,TMR1IE        ;enable interrupt
      bcf   STATUS,RP0         ;Set: STATUS Bit RP0 OFF
      return                   ;Return From Subroutine
;-------------------------------------------------------

subLB
      clrf  subSTATUS          ;Assign:subSTATUS = 0
      movlw LB_L               ;Assign:W = LB_L
      subwf A_L,F              ;Calc: A_L = A_L - W
      btfsc STATUS,Z           ;IF: STATUS Bit Z ON Then
      bsf   subSTATUS,0        ;Set: subSTATUS Bit 0 ON
```

```
      clrw                    ;Assign:W = 0
      btfss STATUS,C          ;IF: STATUS Bit C OFF Then
      movlw 1                 ;Assign:W = 1
      subwf A_H,F             ;Calc: A_H = A_H - W
      btfss STATUS,C          ;IF: STATUS Bit C OFF Then
      bsf   subSTATUS,1       ;Set: subSTATUS Bit 1 ON
      movlw LB_H              ;Assign:W = LB_H
      subwf A_H,F             ;Calc: A_H = A_H - W
      btfss STATUS,C          ;IF: STATUS Bit C OFF Then
      bsf   subSTATUS,1       ;Set: subSTATUS Bit 1 ON
      btfss STATUS,Z          ;IF: STATUS Bit Z OFF Then
      bcf   subSTATUS,0       ;Set: subSTATUS Bit 0 OFF
      return                  ;Return From Subroutine
;--------------------------------------------------------

subRANGE
      clrf  subSTATUS         ;Assign:subSTATUS = 0
      movlw Range_L           ;Assign:W = Range_L
      subwf A_L,F             ;Calc: A_L = A_L - W
      btfsc STATUS,Z          ;IF: STATUS Bit Z ON Then
      bsf   subSTATUS,0       ;Set: subSTATUS Bit 0 ON
      clrw                    ;Assign:W = 0
      btfss STATUS,C          ;IF: STATUS Bit C OFF Then
      movlw 1                 ;Assign:W = 1
      subwf A_H,F             ;Calc: A_H = A_H - W
      btfss STATUS,C          ;IF: STATUS Bit C OFF Then
      bsf   subSTATUS,1       ;Set: subSTATUS Bit 1 ON
      movlw Range_H           ;Assign:W = Range_H
      subwf A_H,F             ;Calc: A_H = A_H - W
      btfss STATUS,C          ;IF: STATUS Bit C OFF Then
      bsf   subSTATUS,1       ;Set: subSTATUS Bit 1 ON
      btfss STATUS,Z          ;IF: STATUS Bit Z OFF Then
      bcf   subSTATUS,0       ;Set: subSTATUS Bit 0 OFF
      return                  ;Return From Subroutine
;--------------------------------------------------------

addHALFRANGE
      movlw hr_L              ;Assign:W = hr_L
      addwf A_L,F             ;Calc: A_L = A_L + W
      btfsc STATUS,C          ;IF: STATUS Bit C ON Then
      incf  A_H,F             ;Calc: A_H = A_H + 1
      movlw hr_H              ;Assign:W = hr_H
      addwf A_H,F             ;Calc: A_H = A_H + W
      return                  ;Return From Subroutine
;--------------------------------------------------------

clearRBIF
      btfsc PORTB,7           ;end mismatch
      bcf   INTCON,RBIF       ;clear int-on-change flag
      bcf   INTCON,RBIF       ;clear int-on-change flag
      return                  ;Return From Subroutine
;--------------------------------------------------------

Store_Value
      bcf   STATUS,IRP        ;set mem bank
      movlw 70                ;starting address
      addwf Valid_Count,W     ;move pointer
```

```
      addwf Valid_Count,W      ;to next 16bit
      movwf FSR                ;set mem address
      movf  A_L,W
      movwf INDF               ;store low byte
      incf  FSR,F              ;move to next byte
      movf  A_H,W
      movwf INDF               ;store high byte
      return                   ;Return From Subroutine
;--------------------------------------------------------

Calc_Average
      clrf  Sum_H              ;clear sum
      clrf  Sum_L              ;Assign:Sum_L = 0
      bcf   STATUS,IRP         ;set mem bank
      movlw 72
      movwf FSR                ;first address
Calc_Loop                      ;Label
      movf  INDF,W             ;load next low byte
      addwf Sum_L,F            ;add it to sum
      btfsc STATUS,C           ;if it carries
      incf  Sum_H,F            ;add to high byte
      incf  FSR,F              ;move to next address
      movf  INDF,W             ;load next high byte
      addwf Sum_H,F            ;add it to sum (won't carry)
      incf  FSR,F              ;move to next address
      movlw 88                 ;Assign:W = 88
      subwf FSR,W              ;is FSR 88 yet?
      btfss STATUS,Z           ;if not
      goto  Calc_Loop          ;GoTo: Calc_Loop
      return                   ;Return From Subroutine
;--------------------------------------------------------

dispmotor
      movf  Sum_L,W
      movwf PA                 ;Assign:PA = Sum_L
      movf  Sum_H,W
      movwf PB                 ;Assign:PB = Sum_H
      clrf  desiredspeed       ;reset desired speed to 0
      btfsc PB,7               ;Is tuning error negative?
      goto  bitneg             ;GoTo: bitneg
bitpos                         ;Label
      movlw 64                 ;Assign:W = 64
      subwf PA,F               ;Calc: PA = PA - W
      clrw                     ;Assign:W = 0
      btfss STATUS,C           ;IF: STATUS Bit C OFF Then
      movlw 1                  ;Assign:W = 1
      subwf PB,F               ;Calc: PB = PB - W
      btfss STATUS,C           ;IF: STATUS Bit C OFF Then
      return                   ;Return From Subroutine
      movlw 30
      movwf desiredspeed       ;Assign:desiredspeed = 30
pos                            ;Label
      movlw 120                ;Assign:W = 120
      subwf desiredspeed,W     ;Calc: W = desiredspeed - W
      btfsc STATUS,Z           ;IF: STATUS Bit Z ON Then
      return                   ;Return From Subroutine
      movlw 30                 ;Assign:W = 30
```

```
      addwf desiredspeed,F    ;Calc: desiredspeed = desiredspeed + W
      movlw 128               ;Assign:W = 128
      subwf PA,F              ;Calc: PA = PA - W
      clrw                    ;Assign:W = 0
      btfss STATUS,C          ;IF: STATUS Bit C OFF Then
      movlw 1                 ;Assign:W = 1
      subwf PB,F              ;Calc: PB = PB - W
      btfss STATUS,C          ;IF: STATUS Bit C OFF Then
      return                  ;Return From Subroutine
      goto  pos               ;GoTo: pos
bitneg                        ;Label
      movlw 64                ;Assign:W = 64
      addwf PA,F              ;Calc: PA = PA + W
      clrw                    ;Assign:W = 0
      btfsc STATUS,C          ;IF: STATUS Bit C ON Then
      movlw 1                 ;Assign:W = 1
      addwf PB,F              ;Calc: PB = PB + W
      btfsc STATUS,C          ;IF: STATUS Bit C ON Then
      return                  ;Return From Subroutine
      movlw 226
      movwf desiredspeed      ;Assign:desiredspeed = 226
neg                           ;Label
      movlw 136               ;Assign:W = 136
      subwf desiredspeed,W    ;Calc: W = desiredspeed - W
      btfsc STATUS,Z          ;IF: STATUS Bit Z ON Then
      return                  ;Return From Subroutine
      movlw 30                ;Assign:W = 30
      subwf desiredspeed,F    ;Calc: desiredspeed = desiredspeed - W
      movlw 128               ;Assign:W = 128
      addwf PA,F              ;Calc: PA = PA + W
      clrw                    ;Assign:W = 0
      btfsc STATUS,C          ;IF: STATUS Bit C ON Then
      movlw 1                 ;Assign:W = 1
      addwf PB,F              ;Calc: PB = PB + W
      btfsc STATUS,C          ;IF: STATUS Bit C ON Then
      return                  ;Return From Subroutine
      goto  neg               ;GoTo: neg
      return                  ;Return From Subroutine
;-------------------------------------------------------

setuppwm
      bsf   STATUS,RP0        ;Set: STATUS Bit RP0 ON
      movlw 255
      movwf PR2               ;Assign:PR2 = 255
      bcf   STATUS,RP0        ;Set: STATUS Bit RP0 OFF
      clrf  CCPR1L            ;Assign:CCPR1L = 0
      bcf   CCP1CON,CCP1X     ;Set: CCP1CON Bit CCP1X OFF
      bcf   CCP1CON,CCP1Y     ;Set: CCP1CON Bit CCP1Y OFF
      bsf   CCP1CON,CCP1M3    ;Set: CCP1CON Bit CCP1M3 ON
      bsf   CCP1CON,CCP1M2    ;Set: CCP1CON Bit CCP1M2 ON
      bsf   STATUS,RP0        ;Set: STATUS Bit RP0 ON
      bcf   TRISB,3           ;Set: TRISB Bit 3 OFF
      bcf   STATUS,RP0        ;Set: STATUS Bit RP0 OFF
      movlw B'00000100'
      movwf T2CON             ;Assign:T2CON = B'00000100'
      return                  ;Return From Subroutine
;-------------------------------------------------------
```

```
desirable
;This subroutine ensures that motor speeds are always changed with
finite acceleration. This reduces interference spikes caused by rapidly
changing motor speeds.
      movf  desiredspeed,W    ;Assign:W = desiredspeed
      xorwf currentspeed,W    ;Calc: W = currentspeed XOR W
      movwf EXOR              ;Assign:EXOR = W
      btfsc STATUS,Z          ;IF: STATUS Bit Z ON Then
      return                  ;Return From Subroutine
      btfsc EXOR,7            ;IF: EXOR Bit 7 ON Then
      goto  Label21           ;GoTo: Label21
      movf  desiredspeed,W    ;Assign:W = desiredspeed
      subwf currentspeed,W    ;Calc: W = currentspeed - W
      movwf EXOR              ;Assign:EXOR = W
      btfsc EXOR,7            ;IF: EXOR Bit 7 ON Then
      goto  Label20           ;GoTo: Label20
      movlw 2                 ;Assign:W = 2
      subwf currentspeed,F    ;Calc: currentspeed = currentspeed - W
      return                  ;Return From Subroutine
Label20                       ;Label
      movlw 2                 ;Assign:W = 2
      addwf currentspeed,F    ;Calc: currentspeed = currentspeed + W
      return                  ;Return From Subroutine
Label21                       ;Label
      movlw 254               ;Assign:W = 254
      btfsc currentspeed,7    ;IF: currentspeed Bit 7 ON Then
      movlw 2                 ;Assign:W = 2
      addwf currentspeed,F    ;Calc: currentspeed = currentspeed + W
      return                  ;Return From Subroutine
;--------------------------------------------------------

makemotorgonow
      btfsc currentspeed,7    ;IF: currentspeed Bit 7 ON Then
      goto  backwards         ;GoTo: backwards
      bcf   PORTB,0           ;Set: PORTB Bit 0 OFF
      movf  currentspeed,W    ;Assign:W = currentspeed
      addwf currentspeed,W    ;Calc: W = currentspeed + W
      movwf CCPR1L            ;Assign:CCPR1L = W
      return                  ;Return From Subroutine
backwards                     ;Label
      bsf   PORTB,0           ;Set: PORTB Bit 0 ON
      movf  currentspeed,W    ;Assign:W = currentspeed
      addwf currentspeed,W    ;Calc: W = currentspeed + W
      movwf CCPR1L            ;Assign:CCPR1L = W
      return                  ;Return From Subroutine
;--------------------------------------------------------
      END
```

Appendix H

Technical Drawings

Exploded View of Mechanical System