Date: June 8, 2024 at 3:42:07 PM EDT
Subject: Design By Constraints

In my previous email I showed how constraints can be used for picking stocks. This is a relatively trivial problem and the hardest part of solving it is not identifying the constraints but gathering the required information. Luckily with modern browsers and free software like JavaScript, along with free information from sites like Yahoo, this is reasonably easy.

I also use constraints for solving crossword puzzles and the like, but its even more useful for design problems.

I learned long ago that the older I get, the more I need to exercise. The only problem is that I'm a big guy and I destroy exercise machines. I went through a series of cheap Chinese machines before I finally bought the a commercial elliptical machine which was rated for use by the Hulk. When I got it home, it took all I had to drag it into the house and assemble it. Mechanically, it's extremely over built. Electronically, it was a piece of crap and the circuit board gave out a puff of smoke and died after a few seconds of use.

I didn't bother trying to fix the board or get a replacement because I intended to hack the machine and connect it to the computer I use for watching videos while I exercise. I like to have a display on the screen showing my exercise statistics while I'm watching a video, and also have the computer control the machine.

The machine is self powered in that the user turns a generator which is connected through a control circuit to an electronic brake. No external power is required and the machine is very quiet when in operation. https://chihua.com.tw/en/brake-fb6/ This is a beautifully elegant solution compared to a powered treadmill which is noisy, requires a lot of power to operate, and just seems wrong when the user generates more than sufficient power to operate the machine and ancillary electronics.

The main problem is that the output of the generator is very extreme in terms of both voltage and frequency and also input power from the user. The generator voltage can easily vary over a 25 to 1 range and the frequency also can range over the same range. The brake current also needs to be controlled over a wide range while maintaining a constant load. On top of all this, because the machine is an elliptical, the generator speed will vary over over one step with the speed dropping significantly at the stroke ends.

The original circuit board was incredibly complex with multiple transformers, power supply ICs, high side current sensors and to top it all off, a lead acid battery presumably used to start the electronics when the machine started.

I hate batteries, they are unreliable, have a short life, are difficult to dispose of and have no reason to be on a self powered machine. The new controller for the elliptical had to be completely self powered, use simple electronics and no battery.

I'm not an electronics engineer and I don't have the mathematical skills to be one, but I do have design skills and with my recent discovery of how to use constraints, came up with a very simple control system.

The main problem is the wide range of generator output voltage and frequency. Typical high efficiency switch mode power supplies can handle an input voltage range of about three to one. They also accept input frequencies from DC to 400 Hz. The frequency range is sufficient, but the voltage range is not even close. Also, the output of the generator is three phase which complicates things a bit more while also simplifying other parts of the design.

My solution is to use an older solution, a phase controlled triac such as used in light dimmers. These work by switching on the triac relative to the phase of the incoming AC. If turned on early relative to the start of the cycle, the output will be high. If the triac is turned on late in the cycle, the output will be low. https://ozuno.com/blog/phase-dimming-a-tutorial/

This is great for single phase circuits with a fixed input frequency, but for the elliptical, the input frequency varies over a ten to one range so simple phase control is not possible. Instead, a simple constraint is used. If the output voltage is too high, the triacs are kept off until the output voltage is too low and then they are turned on.

The power triacs are driven by light triggered triacs which are in turn controlled by an LED. Turning on a triac is done by simply applying current to the LED. The triac will turn itself off when the current through it drops to a low level. Triacs are AC switches and the output has to be rectified and filtered. This is easily done by using a full wave bridge rectifier and a large filter capacitor.

No complex equations are required for specifying the parts because the cost of a high current, high voltage triac is quite low and paradoxically, smaller triacs tend to be more expensive than larger triacs because they are not used as often. I just picked a 600V triac with a current rating well over the expected maximum current. These cost less than a $1 each and the triac gate driver also costs about a $1.

A simple formula is also used to specify the capacitor size. The maximum brake current is 2A which is only possible at high generator speeds. Also, because of the three phase output and bridge rectifier, there is minimal output ripple because there are six voltage peaks for one generator rotation. The brake resistance is 12 Ohms so it requires 24V across it for 2A of current. To guarantee sufficient voltage, the power supply is set to be 30V with peak ripple of 10% or 3V. To get that current the generator has to run at 600 RPM or 100 Hz. The time between voltage peaks is then about 1/6 of 1/100Hz = 1.7 milliseconds.

A rough approximation of capacitor discharge is V = Current X Time / Capacitance. Rearrange to solve for capacitance. Capacitance is Current X Time / Voltage. Required capacitance is 2A X 1.7mS / 3V = 1,133 uF.

There's no need to use more complex formulas because available capacitors have a very limited selection of values and very wide tolerances. This is dirty little secret about electronics design. Most electronic components have extremely wide tolerances and a circuit designed to work within these tolerances doesn't require high precision formulas for its design. Instead, the circuit has to be designed to constrain its output for a given input by using feedback or a particular circuit topology.

Discrete transistors have gains specified over at least a three to one range or even higher. There's no reason to use fancy design equations because these may give very precise results, but require impossible to find parts. Instead, it's better to use a simple feedback circuit with simple constraints. These are very easy to design and require relatively simple rules of thumb. For

example, the current flowing through the base bias circuit should be about ten times the transistor base current.

For the triac control, three transistors are used to control the voltage. Two transistors are used as a differential pair with the output used to drive a third transistor which modulates the LED current to the triac drivers. The transistors cost pennies each and can withstand 60V so even if the power supply ripple is relatively high, the transistors can easily handle the voltage.

Another problem to solve is how to start the circuit. The triac control circuit requires power to turn on the triacs, but unless the triacs are on, there's no power to the circuit.

In the original circuit this was solved by using a battery, but this is a very poor solution and something a little more creative is required. The problem of a very high input voltage range is the biggest issue. Ideally, the circuit should start with a minimum input voltage of 10V and tolerate up to 250V. The input current has to be very low to keep power dissipation low, but at least 15 milliamps are required to drive the LEDs. At 250V and 15 milliamps, the power dissipation is equal to 250V X 15 milliamps = 3.75W.

Although this may seem relatively small, for a human powered machine this is a relatively large value and it also requires high voltage transistors, high wattage resistors and a heatsink. The transistors are cheap, but the power resistors and heatsink are not only large, but also expensive.

Although the triac driver LEDs require 15 mA, they only require this current for a very short time, about 10 microseconds. Also, this current only has to be applied at a relatively low duty cycle because once a triac fires, it can dump a lot of current into the filter capacitor which can power the control circuit for many generator cycles. Once the filter capacitor is charged up, more than sufficient power is available to keep the control circuit going. No batteries required.

The only constraint on the triac control circuit is that there are sufficient pulses to guarantee that the triacs will turn on at some time during a generator cycle. Exactly when is not that critical. A triac may turn on too early or too late, but over the long term, the average voltage will be reasonably well regulated.

The triac drive circuit is implemented using a relaxation oscillator. This is consists of a resistor charging a capacitor. Once the capacitor reaches a certain voltage a transistor turns on and dumps current into the LED until the capacitor discharges to a low enough voltage to shut the circuit down. The capacitor then starts charging again and the cycle repeats.

The charging current can be very small because the time between discharges can be relatively long. During discharge, the capacitor can supply more than sufficient current to light the LED for the short time required. In fact, the capacitor can supply sufficient current to damage the LED and a current limiting circuit is required prevent this.

The capacitor will have two charge paths, an extremely low current path directly from the generator which can work from 10V to 250V and a second path from the 30V supply which will supply sufficient continuous current to keep the triac driver circuit running.

The triac drivers are controlled by simply shorting out the LEDs with a transistor. If the supply voltage is too high, the LEDs are shorted. If the

supply voltage is too low, the short is removed. The current limiter prevents high current through the shorting transistor.

The oscillator circuit uses two transistors with a third transistor used as a simple current limiter.

The circuit design is bit complicated, but is still done using relatively simple circuit theory and capacitor charge discharge equations. Tight tolerances are not required and simple design rules of thumb are used. The only critical constraints are that the current pulse be at least 10 microseconds, the LED current be at least 15mA and at most 50 mA, and at least two pulses per cycle are available at the highest generator RPM.

No high voltage or high power components are required because the capacitor charges only to around 10V before it is discharged. The charging resistor is 180k so the peak power dissipation is 250V * 250V / 180k = 347 milliwatts. The average power dissipation is much lower and a standard 1/4W resistor is sufficient.

No control theory has been used in the design because it's not necessary. As long as there is one part of the circuit which is much slower than other parts and the control itself is generally non-linear, no fancy pants analysis is required.

The technical term for this is a "Bang-Bang" controller. The output is either on or off with no linear region. The traditional way of designing control systems is very difficult with complex equations with lots of poles and zeros, Nyquist diagrams, and Bode plots which are necessary if all parts of the control system operate at the same speed.

But if one part of the control system is substantially faster than all other parts and behaves in a simple on off manner, the design becomes extremely simple. In the elliptical power supply circuit, the triacs cannot switch faster than the generator output frequency which is at most 100Hz while the on-off control circuit can respond at 10,000 times faster and is either on or off. This is why a math challenged person with a very shallow understanding of control theory can easily design this type of circuit. I know just enough to stay out of trouble while still being able to make the circuit work. The trick is to set enough constraints around the problem with each constraint requiring only a minimal amount of precision which is well within my rather limited design abilities.

Although the 30V power supply works great, the brake current still has to be controlled to relatively high precision. Based on my experience with the machine, even small changes in brake current are very noticeable. After the original control board fried, I used a programmable power supply to run the brake temporarily for both research and to be able to use the machine. I was then able to determine the required brake current and voltage.

The brake consists of an electromagnet which is bolted close to the metal flywheel. The movement of the flywheel in the magnetic field of the electromagnet generates electrical currents in the flywheel which then heats up. The combination of power loss in the generator windings and the flywheel heating creates the machines braking effect. Changing the current through the brake changes the load on the generator and also the losses in the flywheel. The control circuit also adds to the losses, but it's power consumption is very low.

The current through the brake coil also generates heat which changes the brake resistance. As the brake heats up, the coil resistance increases and if the brake voltage is kept constant, the overall braking effort is reduced.

This means that the brake current needs to be measured and controlled to keep a constant braking effort and also for setting the desired braking effort.

One way to do this is to use a linear current regulator to keep a constant current through the brake. This is not a very good solution because the losses of a linear regulator are very high and expensive heat sinks and power components are required. Instead, a high speed switching transistor is used to control the brake current. By changing the on to off ratio of the transistor, the brake current can be controlled to very high precision.

When the switching transistor is completely on or completely off, its dissipation is very low. It's only when its switching that appreciable power is consumed. A MOSFET power transistor is used and with a switching time of well under 10 microseconds keeping its power dissipation very low.

An Arduino Nano microcontroller is used to drive the switching transistor and is connected to the computer through and isolated USB connection. Although the microcontroller is relatively fast, the time based pulse width control of the switching transistor and the slow current measurement circuit results in a relatively slow control response and a simple bang-bang control strategy won't work. Instead, a traditional PID or Proportional, Integral, Derivative type control strategy is used.

The control response doesn't need to be fast because it only has to compensate for the change in brake resistance over time. The 30V power supply is quite stable and power supply ripple is felt only as slight vibration from the machine. This means that the control response can be substantially slower than the generator and power supply response to the brake and a very simple control model is possible. For the elliptical control only the proportional and integral parts of PID are used.

The proportional part of the control means that the output brake current is proportional to the difference between the desired brake current and the measured brake current. This is generally a low value as higher values will cause oscillation. This is very evident when using the machine as the effort will increase and decrease during one cycle and is very uncomfortable. The proportional value is tuned by adjusting it until the oscillation is no longer apparent.

Proportional control by itself is not sufficient for good control because there has to be an error between the desired brake current and the actual brake current to get an output. When the proportional value is low because of oscillation problems, this error can be undesirably large. To correct for this, the error value can be integrated over time so that the effective error value increases which in turn changes the brake current to decrease the error to effectively zero. Because the error correction is done very slowly, there is no chance of oscillation because the control circuit remains much, much slower to respond than the power supply.

The only problem with integration is that its value can grow without limit. If the machine is stopped for any length of time, the integration error tends to infinity and will require a long time to move back down to reasonable values when the machine is started. This can be corrected by limiting the maximum integrator value and resetting the value back to zero if it goes too far out of range. This is the strategy that is currently used on the machine. When first starting, the effort is briefly relatively high and then almost immediately drops very low and then gradually works it way up to the correct value and remains there. Once there, the current is controlled to better than 0.1% which is incredible considering the simple control circuitry.

The remaining problem to be solved is measuring the brake current to the required precision. At first this would seem to be a difficult problem, but by putting up various constraints a relatively simple solution is possible. To do this, an ancient but well established technique with a few modern enhancements is used.

The main difficulties with the current measurement is that the current is switched at a relatively high frequency, is in a 30V circuit with lots of ripple and noise and the resistor used to measure the current must have a very small voltage drop across it to reduce the power loss.

The measurement is done using a chopper amplifier which uses a few clever tricks to measure very small voltages impressed on a very large voltage.

The chopper consists of two optically controlled switches which first puts the voltage across the current sensing resistor one way into a transformer and then switches to put the voltage in reverse across the transformer. If the the 30V is pristine DC, a small AC voltage corresponding to the voltage across the resistor is output from the transformer. This voltage is easily amplified by an AC amplifier into a high enough value for the microcontroller to measure and then calculate the current.

The reality is that the output of the transformer will consist of the AC voltage corresponding to the brake current along with the switching waveform and the power supply ripple from the triac switching and the brake current changes itself. Also, the AC amplifier output to the microcontroller will have a DC offset voltage which is significantly large compared to the input voltage.

The first constraint is to simply limit the frequency response of the AC amplifier. The switching frequency is about 15kHz while the chopping frequency is around 244 Hz so limiting the amplifier frequency response will greatly reduce the switching frequency noise.

The second constraint is that the input to the microcontroller is known to be AC. The DC offset voltage is relatively constant over short time intervals and the switching frequency is precisely known. If the measured voltage is multiplied by 1 for half of the chopper waveform and then multiplied by -1 for the other half and the values summed and averaged, the DC offset is eliminated.

This is fine if the measuring and multiplying is done perfectly in phase with input voltage chopper switches, but component tolerances mean that the phase cannot be predicted precisely in advance and more constraints need to be applied.

One constraint is that the chopper frequency is known extremely precisely, well under 0.01% and the correlation of the input chopper signal with the sine and cosine of the chopper frequency can be measured. The resulting sine and cosine values represent the imaginary and real parts of a vector rotating at the chopper frequency with the vector magnitude representing the amplified chopper signal.

The amplitude is easily calculated using Pythagoras' theorem as the square root of the the sum of the squares of the cosine and sine values.

The correlation calculation is done by multiplying the measured voltages by sine and cosine and calculating the average value.

The integral of a sine wave multiplied by itself gives a DC value which varies both as a function of the phase difference between the waveforms and the frequency difference. This is the basis of the renowned FFT. The difference is that the FFT calculates the sine and cosine of all frequencies within the signal starting from DC and the primary signal harmonic to the maximum harmonic as determined by the Nyquist sampling limit.

The fast part of FFT is that is can take advantage of redundancies to reduce the number of multiplications required.

For the chopper, only the primary harmonic is of interest so only a single sine and cosine multiplication is required.

Rather than calculating sine and cosine for each measurement and then doing the average, multiple measurements are first summed and then multiplied using a sine lookup table for each sample. Multiplication is commutative so summing can be done first and then the multiplication done after.

No division to calculate the true average is done because this is simply a scaling operation and is unnecessary. Instead the proportional and integral calculations are done using 32 bit floating point with the values scaled up to match the calculated current output amplitude.

For the elliptical, 32 samples are taken for one cycle of the chopper. The analog to digital converter has a maximum resolution of 10 bits, but the input is biased at half its range to effectively give it a bipolar input of +/- 9 bits. The Arduino can handle 32 bit numbers which means that up to 2^22 samples can be summed. In practice 200 samples are summed for a measurement rate of slightly more than one per second.

A 32 entry table containing cosine values for each sample position is used for both sine and cosine multiplication. The sine values are just the cosine values shifted by 90 degrees.

The correlation calculation and averaging has one huge benefit in that it acts as an extremely narrow band filter centered extremely precisely around the chopper frequency. The effective bandwidth is the inverse of the integration time and is roughly about 200 / 244 = 1.22Hz. This massively reduces noise giving a current measurement stable to better than 0.1%.

In a traditional chopper circuit, a second chopper on the output of the the AC amplifier is used to rectify the AC amplifier output and the resulting signal fed back to the input chopper. This is not as effective as the above procedure because it is sensitive to phase shifts in the AC amplifier, has a much wider noise and chopper harmonic sensitivity and it's much more difficult to apply filtering. The main advantage is that the overall gain is much more tightly controlled.

For the elliptical circuit gain control is not as big a problem as it was when the chopper circuit was first used back in the vacuum tube age. Modern resistors and capacitors are much more stable and an op-amp with feedback has very stable gain. The chopper switches are also far superior to the reed vibrators or neon bulb and cadmium sulfide photoconductors of the past.

The correlation calculations are also done using 32 bit floating point which results in very low calculation errors.

I've measured the current using my DMM over a 45 minute exercise and the current is extremely stable and varies about 1 count in 1200.

The microcontroller itself requires a 9V power supply. A commercially available switching power supply is used to run the microcontroller and is run off the 30V.

Because of the high voltages associated with the generator, an isolated USB adapter is used to connect the microcontroller to computer. These are very cheap and rated to very high isolation voltages.

The microcontroller also controls three line powered fans through solid state relays which run whenever the elliptical operates. A retroreflective sensor is used to sense the movement of the elliptical for counting steps and calculating speed and distance.

The main computer runs a Python script with Tkinter used as the thread manager.

The program paradigm is that virtual machines, which I call workcells, communicate to each other over a virtual network. This makes event handling extremely simple and enhances Tkinter's crippled event handling methods.

The network is peer to peer and decouples the parent-child relationship used for graphical presentation from event handling. A parent-child paradigm for graphical presentation is well proven and easy to use, but for event handling is terrible. Although parent to child and child to parent event handling is common, peer to peer events are also very common. For example, the Arduino will send an even whenever it the step sensor is activated. This event is important to the exercise timer as the timer will stop whenever the elliptical is not moving and reset completely if the elliptical is stopped for more than half an hour. The even is also important to the speed display and the distance display. There is no parent-child relationship between these various displays and no reason for that relationship.

But for Tkinter based event handling, the events have to be passed up to a common parent and then down to the child for processing. This is a major pain in the ass and results in very ugly programs. With the workcell paradigm, the different workcells subscribe to the step status message transmitted from the workcell dedicated to sending and receiving messages from the Arduino. To simplify things even further, there is a dedicated workcell just for the USB connection to the Arduino and another workcell dedicated to managing the Arduino USB to serial protocol.

Each workcell has its own visual component which may or may not be shown as required and each workcell has its own error display which is also shown on demand. This makes error display and handling much simpler.

Also, each workcell is truly autonomous and instantiated anonymously. All communication is done by passing text messages and pointers to data. Each workcell determines how it will respond to an event depending only on its internal state. One workcell cannot call functions within another workcell.

The network is maintained as a single queue which prevents race conditions. As events occur they are put on the queue and the method associated with an event is run atomically before the next event is processed. For simple processes this eliminates the need for locks and semaphores because there are no direct function calls.

For more complex processes, a dedicated guard workcell can be used for controlling access to a critical resource. When a workcell needs access to a resource, it sends a request message to the guard workcell. If the resource is free, the guard workcell sends out a status message containing the name of

the workcell currently using the resource. If the resource is currently in use, and the guard workcell receives a request for the resource, it sends the same status message containing the current workcell using the resource. This way the requesting workcell only has to wait until it receives its name in the status message before using the resource.

When a workcell finishes with the resource, it sends a request message to the guard workcell.

If a workcell hangs up and doesn't relinquish the source, the offending workcell is easily identified by the status message received from the guard workcell.

I've used a very similar programming paradigm with JavaScript for my stock information gathering programs using the queueMicrotask, https://developer.mozilla.org/en-US/docs/Web/API/HTML_DOM_API/Microtask_guide function as the event loop handler. This makes event handling in JavaScript substantially simpler and makes dealing with websites like Yahoo, and SEC.gov much simpler.

I realize that a lot of the above is probably of little use to most people, but my hope is that someone will recognize the value in the above ideas and run with it. I don't claim any originality of the above ideas, they are a composite of research into the stock market, signal processing, measurement techniques and solutions to various problems I've encountered over many years. The main takeaway is that problems in such diverse areas are amenable to a very simple technique for solving them.

There is a "Theory of Constraints" known as a management style, but I would argue that the theory of constraints can be applied to problems in all fields. I've demonstrated its use in circuit design, measurements, crossword and and similar word puzzle solving, it's also useful in solving number puzzles like KenKen and Sudoku where the constraints are explicit.

As far as I can tell AI makes extensive use of constraints and its effectiveness in solving crosswords and puzzles shows that its probably also used by our brains in some way.

The elliptical project is documented at https://sourceforge.net/projects/elliptical-controller/ This is a very niche project, but I'm certain the measurement and control techniques used in the project have lots of other uses. Also, for a lot of circuit design applications, it's not always necessary to use integrated circuits. Sometimes old school is the way to go. The ancients did incredible things with stone knives and bearskins. Imagine what they could do with modern devices. https://www.youtube.com/watch?v=5u-gLR3etLY