Moonwalking Automaton

Our automaton is inspired by a simple concept. We wanted to design and build a simple mechanical puppet that could automatically dance, without prompts or switches, to music being played in the room. Of course, “dance” is a loosely-defined term, so we settled upon a narrowed version of the art. Harkening back to our 80’s childhood memories, we opted to make our robot simulate the motions of Michael Jackson’s famous “Moonwalk” dance, played to the tune of his first big solo debut, the classic title “Thriller.”

1. Construction: Mechanical and Computational Elements

At the highest level, our automaton continuously listens for sound signals. When music begins to play (above a certain volume threshold) the sensor triggers several motors, causing our puppet to perform the Moonwalk. The puppet (artfully displayed as Anubis, the Egyptian god frequently featured in Cabaret Mechanical Theatre productions) automatically starts and stops dancing according to the play of the song. In conjunction with the Moonwalk, a rotating backdrop also glides by in the background, giving our puppet the illusion of gliding backwards during the dance.

To achieve this, nearly all of our mechanisms (both computational and physical) underwent several major design changes and modifications before arriving at a final presentation. The following description illustrates the “final mechanisms” we presented, after many trial-and-error sessions.

Sound Sensor

The only “sensing” mechanism our device uses is a simple Volume Sensor connected to a PicoCricket™ device. When the music starts (as defined below in “Computation”), the PicoCricket triggers three motors and one decorative LED light. Two of the motors power a set of gears connected to a “Rack & Pinion” device, moving Anubis’ feet backwards & forward. Timing is set so that the two legs will alternate their motion back & forth, always opposing each other (dancing in unison). We will refer to these PicoCricket motors as the “dance motors.”

In conjunction with the dancing legs, the PicoCricket also displays an LED light (set above the dancer, as if on the ceiling) that flashes & changes color as the music plays.

The third motor is set above the dancer (placed on top of Anubis’ “box housing”), which powers the rotating backdrop (described below) from above. We will refer to this as the “backdrop motor.”
We will describe each of these components (from start to finish) separately:

**Anubis’ Moonwalk**

This is the key motion of our automaton. The PicoCricket Sound Sensor continuously loops through the main program, recording volume levels as it goes. When several consecutive volume levels (approximately 10ms apart) exceed a pre-defined cutoff, the Cricket triggers the dance motors, which turn on simultaneously, each of which turn a single gear, powering a “rack and pinion” drive mechanism.

The mannequin movement was derived in part from a limited joint which allows 180 degrees of movement, and locks in a straight leg position, preventing hyperextension of the knee. The other feature that helps produce the walking movement is a vertical slot in hip joint; the top of the leg moves through this slot. When the leg is at the front end of its stride, it locks, and when pushed backwards, it is allowed to stay locked by moving upwards in the hip slot. At the back end of the stride, the weight of the knee joint causes the knee to bend and it stays bent through the forward movement, remaining at the bottom of the hip slot. The body parts were modeled on the laser printer using Corel Draw. Joints were connected using nuts and bolts loosely fitted into the holes in the wooden pieces, with the nuts and bolts were soldered together to prevent slippage.

The box housing the mannequin was constructed using simple brackets, and serves to suspend the upper body of the mannequin “in mid-air,” which is necessary in order to achieve the desired effect. The box is raised at the base for the racks (of the rack and pinion system, described below) to move through without interference.

The rack and pinion system was modeled on Lego gear components and dimensions. For the pinion, we used pre-existing Lego gears for ease of attachment to the Pico motor. The rack was modeled out of wood on the laser cutter to fit the pinion, and consisted of about six inches of teeth with an additional six inch extension attached to the feet of the mannequin. The rack with extension was then laid in a recessed track made from layers of wood. We used a separate rack, pinion, and motor for each leg to allow for complimentary movement in opposite directions.
LED Light

The simple (and most trivial) component of our automaton is a small LED light, controlled by the PicoCricket. When the PicoCricket triggers the motors, it simultaneously begins a background process that displays a color-changing LED pattern in conjunction with the motorized movements. The LED is placed on the “ceiling” of Anubis’ box, making a crude approximation of eighties-era dance-club lighting.

Rotating Backdrop

The third motor is set atop Anubis’ box to rotate the backdrop. Our backdrop is a classic “Thriller” promotional poster, printed (via a large-format photo-printer) on the back of a flexible-yet-strong sheet of white Tyvek. This continuous circular backdrop is fed over two ½”x16” wooden dowels that rotate in a single direction (backwards) in conjunction with the dancing of Anubis’ feet.

The motion of this mechanism is very straightforward (it simply rotates one of the dowels), but in order to overcome the large amount of friction in this device, we couldn’t directly rely on the PicoCricket motors (which were far too weak) to
do the job. Even after sanding the components and using ample graphite, it required a more powerful motor. Instead, as the PicoCricket read volume levels, it fed those values to a variable-power LED light, which is taped directly to a photon-sensor on a HandyCricket™ Board. The HandyCricket reads these simulated “Photon-Volume readings,” and when they exceed a threshold (corresponding with the same threshold as the dance motors), the HandyCricket turns on a single motor port. This motor port doesn’t directly power the motor, however. Rather, it turns on an electrical relay switch, which has been soldered to feed a much-more-powerful 12-volt power supply to a HandyCricket motor (which has also been soldered in place for the same purpose). Fed directly from a wall-plugged power-supply (no batteries necessary), this 12 V supply drives the motor extremely fast (~1000-2000 rpm). We “gear-down” the motor by a factor of 50x using a series of Lego-gears strapped atop the box, giving a rubber belt-drive (rendered from a small plumber’s O-ring) sufficient torque to rotate the high-friction backdrop in a continuous loop. The end-motion is relatively slow but very powerful, and such a mechanism could feasibly be used to indirectly power all sorts of high-friction devices using the relatively low-powered outputs of the Cricket motors.

**Computation**

The “Computational Element” of our Automaton rests in two devices. The PicoCricket does the primary computations. It continuously reads from the Volume Sensor, remembering a small series of consecutive readings. When those readings all surpass a threshold (thus avoiding accidental starting by ambient “noise”), it triggers three devices simultaneously. The first is a motor-port… it sets the power and duration of two Cricket Motors to directly correspond with a “beat” in the Thriller song. The second device is the variable-power LED feeding light-sensor readings into the HandyCricket. The third device is an LED that plays a continuous loop of variable-color patterns onto the display. This final LED process runs in the background, so as not to interfere with the continuous looping of noise-level readings on the PicoCricket. The HandyCricket is only responsible for detecting Photon-sensor readings from the PicoCricketLED, and turning on a relay switch accordingly, which (as explained above) powers the rotating backdrop.

**2. Implementation: What Worked and What Didn't**

As mentioned previously, almost our entire model above was finished after trying (and discarding) several other attempts. We already outlined the working demo version of our automaton above… we will now try to outline the breadth of our other efforts here:
Part of our original goal was to create very lifelike motion in the automaton’s moonwalk. At the start of the project we sketched out diagrams and notes on how to accurately recreate the leg and foot kinematics of this technically challenging foot glide, using online videos of Michael Jackson performances combined with our own past experiences and knowledge of the technique. Most of the mechanical design efforts focused on fulfilling the resulting constraints of this initial design concept.

After much conceptual exploration and analysis, we decided to separate the vertical and horizontal components of the foot motion into different synchronized mechanisms of cyclic force transfer. Once generated, these two motions were to be linked back together in guiding the movement of the dancing figure’s feet.

We identified the defining vertical component of accurately simulating a moonwalk as the heel motion, which needs to have a smooth lift and a snappy drop. We carefully estimated the time curves of the relative heel height based on our investigations of moonwalk kinematics, and mapped the resulting transitions into the curve of a cam shape, such that it’s intersection of it’s edge with a horizontally placed vertical line recreated the original curves. This was then used to draft the dimensions for a pair of cams and matching vertical slider, with two downward projections that move vertically thru fixed horizontal guide slots in the structure of the automaton’s base. The cam pushes the bottoms of these projections produce the proper vertical motion, and a parallel pair of guide slots above transfers vertical lift while accommodating the free horizontal motion of a pin associated with one of the model’s feet.

We identified the horizontal component of a moonwalk as a simple back and forth motion of the foot at a constant speed. Even though the motion is simpler than the vertical component, the requirement of a saw-tooth rather than sinusoidal time curves of the back and forth motion presented us with a technical challenge. The rotary motions of such components have a sinusoidal back and forth projection along any one dimension, and a linear speed unidirectional tangential force. To create a saw tooth motion required that we find some clever way of combining these two elements.

Our initial solution for meeting these horizontal motion constraints was to combine a small length of chain with two very small sprockets, and affix a laterally projecting pin from the side of one chain link to guide the motion of a horizontal slider. The chain link rolls along at a constant velocity from the top of one sprocket to the other, and then reverses directions while maintaining speed as it passes from the bottom of the second link back to the first. The horizontal slider was designed to ride along two rails, and incorporates a guide slot to transfer horizontal alignment while allowing free vertical motion of the same pin associated with the model’s foot that is attached to the vertical slider. With small enough sprockets (relative to the chain length), this results in a fairly good approximation of a saw-tooth motion, similar to a human’s feet relative to their center of gravity during smooth walking.
After much tangling with complexity of designing and arranging the shared fit of these mechanisms into the same casing, the vertical and horizontal components of motion were both fully and successfully implemented before the project presentation & demonstration deadline.

However, their full integration still presented two technical challenges that were not fully resolved by that time. One was the design of the foot linkage mechanism that would combine and use the motion generated by the horizontal and vertical sliders to recreate the desired foot motion. The other was to drive these mechanisms with appropriate force and synchronized speed. Both of these tasks had been initially and mistakenly perceived as trivial, and so there was quite a rush in the last week and a half when it was discovered that they required significant work to resolve.

First, and perhaps most critically, was incorporating the foot linkage assembly. Our initial idea, for animating a foot with a rotating joint at the ball of the foot, was to put a hidden extension on the heel part of the foot that would extend beneath the “floor” of the display and act as a lever for actuating the heel lift and a point of force for transferring horizontal motion. On closer inspection, it turned out that the behavior of the lever in our
initial simple approach would not accurately transfer either the vertical or horizontal motion of the guide pin below, whose precise paths we had so painstakingly crafted.

The first problem was that the lever action the foot extension acted as a required a reversal of the vertical component of motion, and introduced aberrant horizontal displacements of the foot relative to the guide pin (which already had the correct horizontal and vertical motion).

The vertical component was easily corrected. Because the foot lever reversed the direction of force applied to it, we needed to make the cam push the vertical slider down, rather than up. This was accomplished by introducing a slot in the slider thru which the cam could rotate and exert downward force on the lower end of. We also introduces small attachment points to the slide and frame pieces to replace the downward counter force of gravity with the upward counter force of a rubber band.

Correcting the horizontal component of the foot motion required a slightly more sophisticated solution. The vertical displacements of the guide pin at the each end of the figure’s stride introduced corresponding but unwanted horizontal displacements via the heel lever’s rotational transitions between horizontal and vertical orientations. The solution required introducing additional pieces to the foot assembly to counter act the horizontal displacement while maintaining horizontal force transfer. The first of these was a second lever segment to indirectly connect the heel lift lever to the foot guide pin, with the same distance between attachment points as in the heel lift lever. The other key piece added was a hidden extension projecting beneath the display “floor”, this time from the toe, with a guide slot to keep the two lever segments in vertical alignment. The horizontal displacement of this second lever segment exactly counters that of the segment directly connected to the foot, enabling the accurate direct transfer of horizontal motion from the guide pin to the foot, independent of the guide pin’s vertical motion.

This design concept for correctly transferring the horizontal and vertical motions of the guide pin to the foot was formulated shortly before the project deadline, and was the last key piece for making the automata work correctly. Unfortunately generating a complete design for this new assembly would require time and a careful skilled effort, and no empirical proof existed that it would work on the first try.

Given our time constraints, and the temptation of a much simpler, though less functional, design idea, the team voted at the last minute in a 2-1 split to forego pursuing the implementation of the new foot linkage that would complete the original design. The bulk of the semester’s mechanical design work was discarded, and instead we ended up using a simple rack-and-pinion device to drive the feet instead.

The other remaining technical challenge of the original design was inherited in a slightly modified form by the new approach: How to power and synchronize the mechanisms of our automata, given the limited power of the cricket motors. The new foot motions were much easier to realize with the rack and pinion design, which required little force (not having to combine the resistance of driving and integrating mechanisms for two
dimensions of force and motion). However, implementing the rolling backdrop of the display forced us to address the motor power and gearing problems anyway.

Although the simpler demo solution didn’t simulate the up & down heel movements inherent in a true Moonwalk dance, it did provide the back-and-forth motions that form the primary component of the foot and leg motion. The elegant anatomical design of the Anubis doll was able to provide a naturalistic result for many of the remaining dance movements, with the knees locking and bending at precisely the right times in the dance.

Extended Dance Movements
Since dancing is not limited to a Moonwalk motion, we originally intended for Anubis to partake in other movements too. Anubis’ arms are currently as flexible as his legs, and we’d looked at using a marionette-style string system to move his arms in beat with the music. We didn’t encounter any constraints to this idea except time, which we ran short of in the end. Given another few weeks, we very-well may have gotten that working.

Beat-Detection
The original version of the Cricket program was written to incorporate a “beat” detection algorithm, which would power the motors faster/slower in conjunction with the music, no matter how fast or slow (or loud or soft) the music was playing. We quickly learned, however, that the PicoCricket “Scratch” software is extremely limited in data-storage. With only seven (7) single variables at our disposal, it was near-impossible to collect data to implement anything approaching “beat-detection.” So instead, we opted to use Handy Crickets, an older (but more computationally powerful and flexible) predecessor of the Pico Crickets. The Handy Crickets could store up to 4K of data memory, approximately enough for our needs.

However, the Handy Crickets had some serious drawbacks, the first being that they have no sound-sensor capabilities; a crucial component for our automata design. After researching the technical specs of Handy Cricket Sensor Ports, we worked considerable hours towards several sound-sensing approaches. The first involved using a small Radio Shack microphone to produce a volume voltage, which could be fed into the Handy Cricket sensor port. We quickly wired up a working component to the oscilloscope, but the output was far too weak for our purposes (0-0.2 V, as opposed to the required 0-5 V).
Using a breadboard and an amplifier chip, we tried wiring together a 25x amplifier for the output signal, but after several days’ effort, it was to no avail.

Instead, we opted to use the Pico-Cricket sound sensor, and “feed” signals into a Handy Cricket using an outside-the-box medium: light. The Pico Crickets have variable-strength LED actuators, and the Handy Crickets come with accurate photon-sensors. Combining the PicoCricket LED with the HandyCricket Photon Sensor, taped together in a simulated “black box,” and tweaking the sensitivity & power of each in the software, we could accurately feed the sound levels pseudo-directly into the Handy Crickets, where the computations could be performed.

Unfortunately “beat detection,” even in its simplest form, is a relatively complex problem, and the slow clock-speed of the Crickets severely limited our real-time sensing capabilities. Using the simplest detection algorithm we could implement, we still were only able to execute about 10 program loops per second… far too slow to take reliable readings of fast-paced pop music.

Also, upon analysis of the sound-detection readings, it became obvious that one of our presumptions was flawed. We’d assumed that a typical song’s “beat” would be measurably (albeit briefly) louder than the rest of a song. This isn’t always so… in fact, using Volume sensors only (with no capabilities for frequency analysis), it proved almost impossible to electronically distinguish a beat from the rest of a loud rhythm, especially in Michael Jackson’s “Thriller.” And hence, beat detection was discarded. Instead, we opted for more-rudimentary (albeit less-flexible) beat-simulation, following “Thriller’s” precise half-second rhythms. In the end, the illusion worked.

3. Context: Influences, Educational/Artistic Aspects, Purpose

The purpose of our project was simple: It’s fun to watch a robot dance. 😊 And let’s face it; anyone who was young during the eighties has tried (on several occasions) to Moonwalk in front of a floor-length mirror. Most folks aren’t very good at it. It’d be cool to build a robot that was.

Much of our mechanical inspiration & “style” was derived from the Cabaret Mechanical Theatre, whose automata frequently animate figures in repetitive and entertaining motions. We chose to fully display our mechanical devices, for any onlooker to view & analyze at their heart’s content. Even our character, Anubis, was lifted directly from their frequently inspired productions.

In the end, we were rather proud of the fact that our Anubis doll could inherently simulate organic leg movements (knee bends, etc) given only the simple back & forth mechanism of our motors. This was no small mistake. We owe much of the final project’s success to the initial elegant design of the doll.
We do (of course) realize that our light-hearted automaton will probably never affect humanity in significant, life-changing ways. It’s a fun toy, little more. But given an actual working beat-detection algorithm and a broader array of possible movements, it could easily be an entertaining exhibit in any hall, auditorium and/or dance floor where noise and music are common ingredients.

4. Improvements: Short-Term, Long-Term

Our biggest possible improvement, given a short-term timeline, would be to fully implement the kinematics of the moonwalk mechanism as we’d initially designed it. That mechanism was very close to finished, and we could have worked out the bugs and finished its construction given a short time longer of focused work. But alas, c’est la vie.

Ironically the design for the foot linkage mechanism that would complete the original design (sans gearing work, implemented anyway) was completed the evening after the demo, and implemented and tested shortly thereafter. It worked beautifully the first time.

In addition, in the week following the demo, several designs for integrating proper coordination of all the mechanisms were implemented and tested, before arriving at a purely geared approach that successfully minimizes friction and supports the full set of movement requirements.

At the time of this writing, the full implementation of an improved design over the original set of physical
mechanisms is in its final stages, with a clear path towards easy integration of additional synchronized body motions. The most tempting and accessible short-term goal is the incorporation of synchronized arm motions and head bobbing.

Short-term computational enhancements are also very tempting. Given a more-powerful computational element (either a faster-running Cricket or perhaps feeding signals into a laptop computer) with a higher sound sampling frequency, we may very-well have been able to get true beat-detection working in short order. Using FFT’s (similar to an oscilloscope’s readings), detecting the “white noise” of a drumbeat (different from the frequency signals of other instruments) would have been a much simpler exercise. Other approaches may also work. Once beats are known, the act of speeding-up or slowing-down movement accordingly is a trivial task in the Cricket software, easily “tweaked” in a couple days’ experimentation.

Lastly, with one more Pico Cricket and a data-connector between (with a few pulley-gears and strings) we may have been able to incorporate arm movement mechanisms into the moonwalking Anubis, giving him a more realistic full-bodied range of movements.

In the longer-term, it could be interesting to incorporate a far wider range of motion, allowing Anubis to (among other things) tilt his head, turn his waist, and freely rotate his shoulders & hips in 3 dimensions. This would, of course, greatly complicate the mechanisms and algorithms involved in generating motion.

Another interesting mechanical possibility is the redesign of the motion mechanisms to fit inside a small flat mobile base, so that when Anubis moonwalks he actually travels
backwards (on hidden wheels) in proportion to the lengths of his strides. This could really enhance the delight at witnessing this the automaton in action.

Instead of Anubis being suspended inside a 4-sided housing box, we’d earlier envisioned the idea of his figure being freestanding, with all the “drive” mechanisms hidden below the platform. Upper-body movements could (in theory) be driven using tension-wires fed through the underside of the box up through the body. The possibilities are endless, but so is the inherent mechanical complexity. Building such devices reliably could easily strain even the most skilled mechanical designers. We had enough difficulty simply getting a hanging Anubis to move his legs correctly; a humbling realization in the face of a wide range of exponentially more-complicated 3-dimensional “freeform” movements. Still, it’s fun to imagine, and given enough time & patience, it may be entirely possible to pull off.
5. Appendix: Source Code, Designs, and Technical Notes

5.1 Initial Design Sketches:

---

**Moon Walk Kinematics**

Hip Height Constant

(“Floats” body)

- Constant distance
- Toe stay down
- Foot Motion
  - approach constant speed
  - approach instant transition
  - eliminate hip motion pause!

---

**Motional Qualities**

- Hip Speed approx. Const.
- “This is called “Travel”
- “Driven” By Foot Motion
  (unless in illusion, like antenna)

---

**Mechanism Concept Art**

2/13/07

- Vertical Slide Component
  - 1
5.2 Laser-Cut Outlines:

Anubis Doll Pieces

Chain & Sprocket Mechanism Pieces as Implemented at Demo Time:
Articulated Foot Mechanism:

Final Working Spec
(with Full Foot Kinematics, Synchronized Backdrop, and New Rack and Pinion Design):
5.3 Code: Handy Cricket (CricketLOGO) device:

```logo
global [
    light  ;; Light sensor reading
    lastlight  ;; Previous Light Sensor reading, for beat detection
    threshold  ;; Light sensor "beat" threshold
    opower  ;; Output power level
]

to main
    ;; Clear the data array, Initialize Variables
    setthreshold 165
    ;; Begin infinite looping
    recurse_main
end

to recurse_main
    ;; Read Sound readings from light sensor
    ;; Set "light" to a lower-is-lower metric, 0-255
    setlastlight light
    setlight ( 255 - sensora ) ;; "Beat" detection
    ifelse ( ( light > threshold ) and ( lastlight > threshold ) )
        [ setpower 8
            b, onfor 10
            ab, onfor opower
        ]
        [ b, brake
            setopower 0
        ]
    ;; Recursive call (interpreted as infinite loop by
```
### in code:

```plaintext
global: V3

block BRAKESTOP
  brake
end
```

---

### Pico Cricket (SCRATCH) device: