Kinetic Sand Table

Authors:
Elisa Bianchi, Emilien Duc, Tugdual Kerjan, Antoine Moix, Elior Papiernik

Supervisor:
Federico Stella

Professor:
Christoph Koch

June 3, 2022

Contents

1 Introduction

This documents presents the *Kinetic Sand Table* project and explains the different steps required to realise the project from scratch.

2 Material Needs

To carry-on with this project, and be sure you will be able to build it without any problem, here is a list of the material you will need :

- 2 x Nema-17 stepper motors
- 1 x Rail system
- 8 x 30cm of 10mm threaded shaft
- 2.5m^2 of 18mm plywood
- 16 3M screws and nuts
- 20 5M screws and nuts
- 40 10M screws and nuts
- 1 sufficiently strong magnet
- 1 steel ball bearing to be used to draw
- 500g sand for chinchillas
- 4 8mm Ball bearings
- Access to a 3D printer and printing material (PLA or PETG)
- Access to a CNC machine
- Access to other basic tools (sanding, drilling, etc...)

3 Hardware

Let's start off with the hardware side of this project. It is composed of two main elements : building the actual table, and the rails/motor mechanisms.

3.1 CNC Table

For the table design, we chose something that would be modular in regard to the placements of the motors, as well as accessible to ensure that maintenance would be hassle-free.

The final table dimension are 65 x 30 cm, respectively the diameter and the height.

The 3D model depicting the table's end result we are going for can be found on our Github repository. This design is comprised of two parts, the top one is an enclosed space meant to contain the sand, as well as the metal ball used to 'draw'. The bottom is fully dedicated to storing the mechanisms and all electronic components.

To start building the table, get your 18mm plywood and your favourite CNC machine, and cut the pieces according to the 3D model as depicted in figure 2 and 3. They can be found on the Github repo as well.

Side note : the set of CNC instructions might need to be adjusted according to your own CNC-setup.

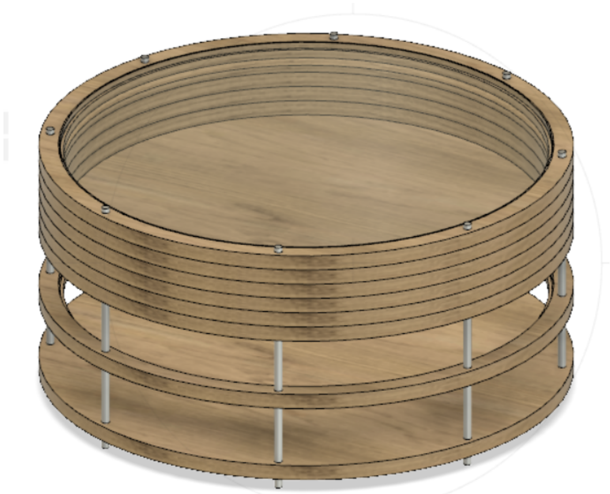

Figure 1: Table Design

Figure 2: Table Pieces CNC

Once you have all the wooden pieces, sand them and set them aside.

You can now grab the threaded shaft, and using a handsaw, cut them into pieces of length 30cm.

You can now use lots of 10M bolt nuts to assemble the table. Use a phone's level app, or an actual air bubble level to check that your table is levelled, and if it is not adjust it by playing around with the bolt nuts.

Be mindful to keep aside 4 pieces with the inner cutout, as these will be used as support for the railing. Note tat the model contains one layer of pieces with a gradual cutout so that you can stick leds on them, facing the center of the table. If you do decide to include this layer, remember to drill an additional hole in all the layers below to accommodate the LEDs' power cable. A picutre of the table with the leds can be seen in ??.

Figure 3: Table Pieces CNC

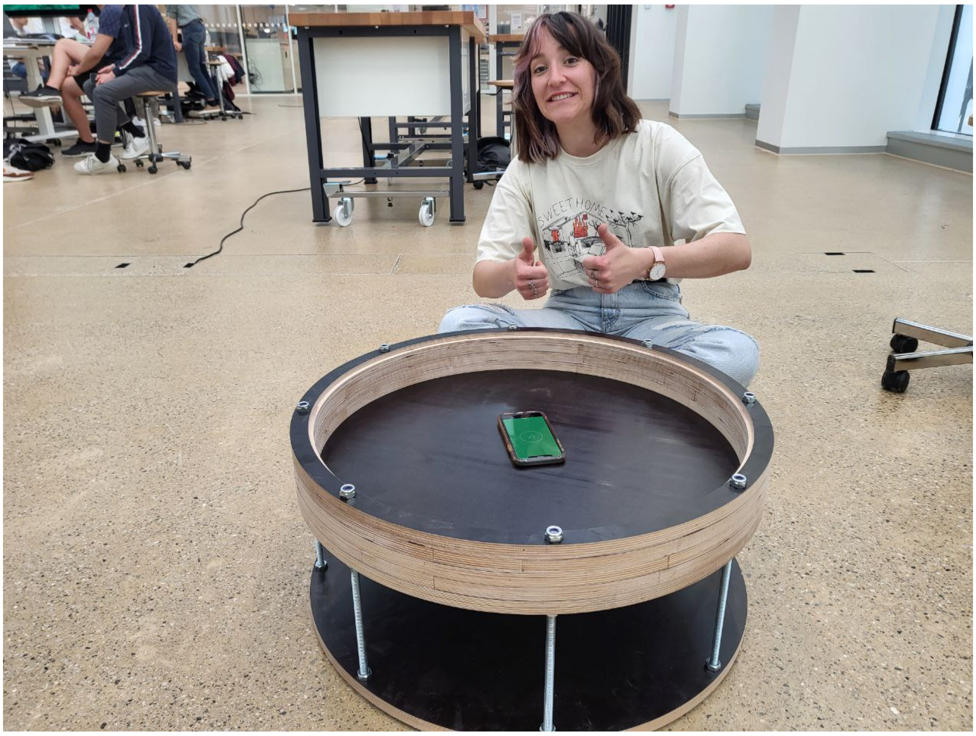

Figure 4: Assembled Table

3.2 Mounting the Rotating Axis

Now that we have a table, let's move on to the hardware that allows us to actually move a metal sphere around.

We start by building a first classic railing mechanism, and mounting it on a 6 mm thick wood plaque, with dimensions 16 x 48 cm.

To ensure that everything is straight, 3D print some supports that will flush everything to the same plane, and then mount it onto the aforementioned plaque, as depicted in fig. 5. You can find the 3D pieces we printed on the Github repository, and adapt them directly on the fusion project in case you need to. Note that the 3D piece from fig. 6 is meant to be used as a captive nut so that the motor shaft is a tight fit with the rotating axis. It is better to glue the nut so that it does move over time. Make also sure that the screw you put in the captive nut comes pushing against the straight side of the motor shaft (and not on a curvy part).

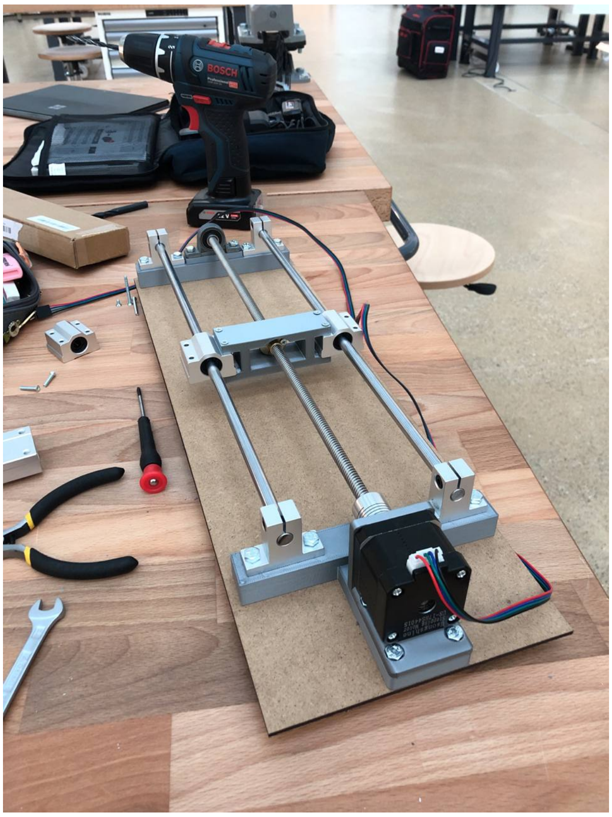

Figure 5: Assembled Railing

As our project aims at building a round table, we will be working in polar coordinates instead of Cartesian ones, and hence we need a rotating factor in our mechanism, which is where the second motor comes into play. The idea is to mount the first platform with the railing system on top of the second motor, which allows us to have a rotation, as well as a back and forth movement, representing respectively the angle θ and the radius r in our polar system.

3.3 Wheels and rails

Since the magnet will move along the axis, the weight balance of the axis is unstable and changes over time. To account for this, we added a pair of wheels on each side of

Figure 6: Captive nut to fix the axis on the motor below

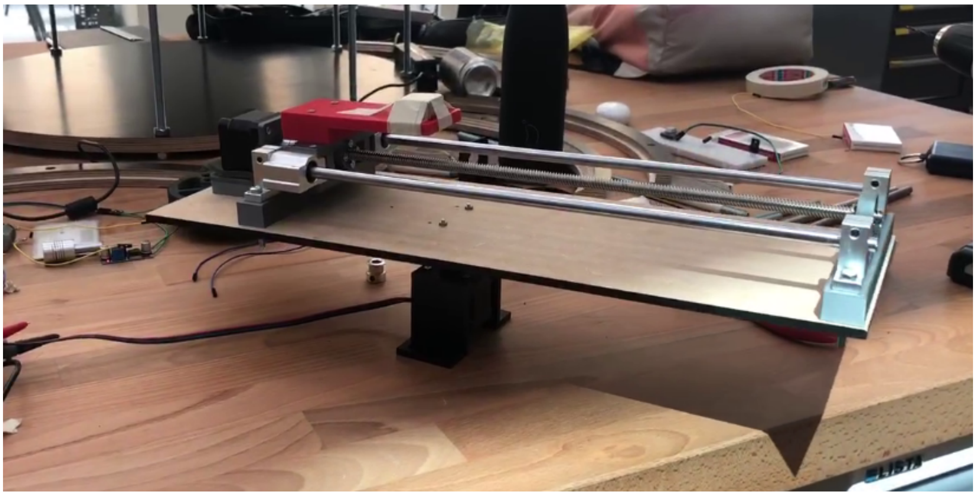

Figure 7: Second Motor Assembly

the rotating axis. The wheels are 4 prints of the same piece, which can be found on the repository. Once you fix them one in each corner, we need to make the necessary adjustments so that each wheel is free to roll and is not too far or too close from the side and rail. Since those adjustments depend on many factors and need to be very precise, we found that the best way to adjust your setting ”on demand” is to heat the arm of the wheel so that you can move it freely and put it exactly where it needs to be. An example of how the wheels are placed in the final setting can be seen in fig. 8

3.4 The ball

At first we used a pretty big magnetic steel ball (not unlike a Geomag one) but the issue was that it was too strong and was pulling the y-axis arm underneath, bending the motor shaft and making the move of the arm impossible.

We then switched to small non-magnetic steel ball bearings, but then the issue was that they sometimes lost attraction to the magnet. We solved this issue by replacing the

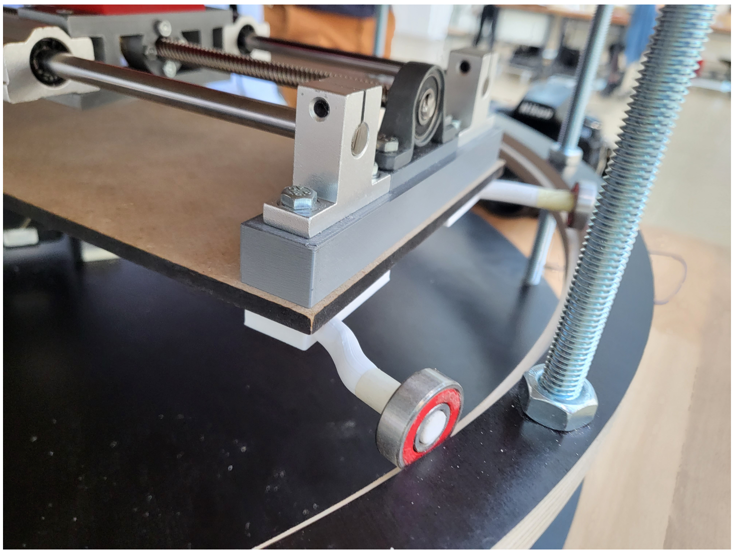

Figure 8: Wheels adjusted to fit on the rail

plate on which the sand stands by a much thinner one.

3.5 Magnet

We first decided to use an electromagnet, but it was unfortunately too weak.

We then used some strong neodymium magnets that were laying around for attracting the ball and moving it around. They were very strong, too strong in fact when used with the magnetic ball, but by adjusting the thickness of the plate, and the distance between the plate and the magnet, we were able to get a nice strong, but not too strong, attraction between the ball and the magnet arm.

3.6 Choice of sand

For the sand, we thought that regular beach sand was too coarse to be effectively drawn into by such a small ball, so we first tried with a mixture of polenta and corn starch, both of which were found in the nearest Migros.

We tried several ratios of theses two elements:

- The corn starch only by itself was too fine and sticky. It also generated a lot of static charges that were slowing the steel bearing.

- The polenta only by itself was too coarse, drawing couldn't been seen clearly and it had a tendency to crumble on itself, ruining the line that was just made.

- With 50/50 polenta/corn starch, it was better, the white color of the starch made drawings much more visible on the black table underside, but the mix was still too dense for the ball to roll through.

- After a while, we found that a ration of 30% corn starch and 70% polenta was better, but still not ideal, so we thought about switching to another sand alternative completely.

- Ultimately, after looking for existing projects online, we finally found the perfect thing: sand for chinchillas!

Chinchillas are some cute pet mammals that come from Latin America. They need a special type of sand to bath in and clean themselves. This sand is very fine and light, and can fortunately be bought quite cheaply on Swiss online pet stores. So we ordered a bag of 5kg of it for less than 10 CHF. Let us tell you, it was worth it! This sand is exactly was we were looking for: it is easy for the ball to roll through, has nice contrast against the black underside of the table, and is able to hold itself in shape and not ruining the drawing.

4 Electronics

4.1 Motors and drivers

We used two A4988 stepper motor drivers to run two NEMA-17-like stepper motors. Each motor is used to move the axis. One is for the circular x-axis and the other one is for the linear y-axis.

Using stepper motors allows us to be very precise in movement, at the cost of an increased complexity to drive them. That's where the Arduino board and the stepper drivers come into play.

One issue we had, is that these cheap drivers are really easy to fry if you are not careful enough. In fact we had to rebuild the electronics part a lot because of how frequent they were destroying themselves. That has slowed us a lot, we spent many hours investigating why the motor would not turn and discover after some sleepless nights that the motor driver was, once again, malfunctioning.

Also, because they are using a much higher voltage than the Arduino, we are forced to used a separate power supply, providing 12V, where as the Arduino is powered through USB.

We had to take some measurement to calibrate the axis. Otherwise, one centimeter in the desired pattern would not account to one centimeter in real life. We also experimented with and without micro-stepping, and found that 1/8th of step was a sufficient resolution and doesn't compromise the maximum speed.

4.2 Arduino and computer link

We used an Arduino Uno board to send instructions to the steppers motors. The Arduino runs grbl, which is a no-compromise, high performance, low cost alternative to parallel-port-based motion control for CNC milling.

Then, the Arduino is connected by USB to a computer and establish a serial communication link that is used to configure and send g-code. That means that the actual sand pattern processing is done on a computer and the Arduino is only used to drive the motors.

We thought about using a Raspberry Pi as the computer running the Telegram Bot, but because of worldwide electronics chips shortage, we were not able to obtain a board in an reasonable amount of time and money. For now, an ordinary laptop is used instead, but the principles are the same. More information on the Telegram bot in the Software section. See the following diagram for an overview.

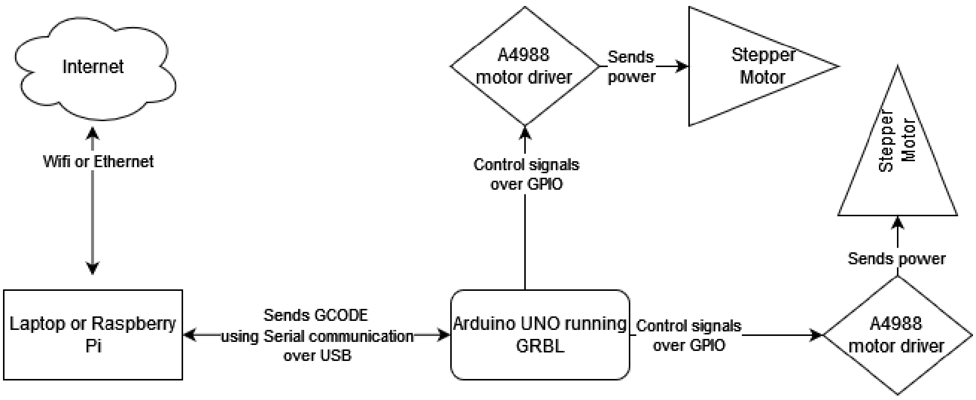

Figure 9: Schematic of the link between the computer, the Internet and the Arduino

5 Software

5.1 Patterns

Typical CNC machines will use g-code as input. The main issue here is being able to convert g-code patterns from Cartesian coordinates to polar coordinates, and communicate these to the board

The issue with grbl is that it does not support polar axis, so a coordinates conversion is to be made on-the-fly before sending it to the Arduino.

A useful site we used initially was Sandify, a tool that can generate different g-code but the output was hard to use and convert to use with our polar machine.

5.2 Interface

So as to have a smooth interface with the user we decided to use Telegram as the interface for the user. Available python libraries allow us to create a bot with whom the user will interact, for example to select the next pattern. The code for this bot can be found on the corresponding folder of our repository. The Telegram bot can be launched with `main.py`.

5.3 Program

Once the pattern that a user wishes to display is selected, the computer running the Telegram bot generates the corresponding g-code and send it to the Arduino via USB. The Arduino will then draw the figure on the sand.

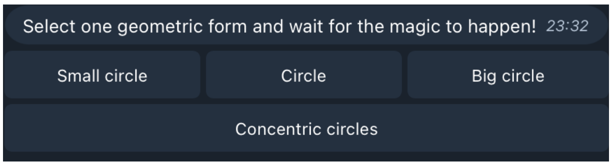

Figure 10: Telegram bot interface. The user can select predefined patterns to be drawn or choose to draw concentric circles

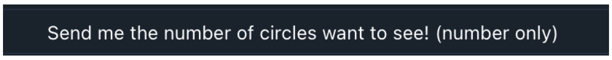

Figure 11: Upon selecting the concentric circles option the user is notified that they should send a number to select the amount of circles

Figure 12: The user can then send a number (only) to choose the amount of circles to draw

5.4 Functionalities

We wanted to develop a way to draw any 2D design but we did not achieve to do it by the end of the course. Specific design can be easier to add if the GCode is generated accordingly.

6 Conclusion

In this project we built a sand table plotter that can be controlled with a Telegram bot. With hindsight, choosing to build a round table was a huge challenge compared to a squared one. Having a functional rotating axis took time and the multiple technical issues we encountered were a huge barrier to our progress. However, with this functional basis, we have a layout that is ready to welcome limitless software designs to draw in the sand. The Telegram bot is a gateway to an easy connection with the table and could be directly executed on an embedded raspberry pi without any adaptation required.

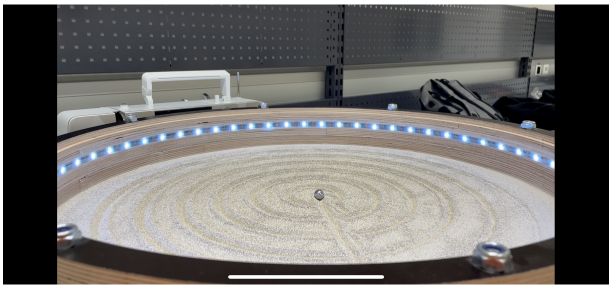

Figure 13: Final result after the drawing of concentric circles

7 Ressources used

- https://lastminuteengineers.com/a4988-stepper-motor-driver-arduino-tutorial/
- https://www.instructables.com/DIY-3-Axis-Polar-CNC-Machine/
- https://www.instructables.com/Make-Music-With-Stepper-Motors/