Combination Lock Solver

Riley Ellis

MMAE 432

***Abstract*—This report presents the design, testing, and evaluation process of a novel device, the Combination Lock Solver. The device, designed to crack combination locks in less than ten minutes, relies on innovative techniques for detecting resistance points to identify the lock's combination. Four potential solutions were explored for resistance point detection, with a dual-shafted stepper motor and an optical encoder emerging as the most effective. The device also utilizes a stepper motor for the lock-solving function due to its precise control capabilities. Feedback mechanisms for verifying the lock's state were also investigated, with an analog feedback servo demonstrating promising results. Despite the initial successes, the device currently has limitations, including inconsistent first combination number determination, solving time exceeding the targeted ten minutes, and physical design constraints. Future development will focus on enhancing resistance detection, improving the lock-solving function, and refining the overall device design. Once these issues are addressed, the Combination Lock Solver has the potential to become a viable product in the market.**

I. Introduction

Combination locks are frequently purchased to secure belongings in lockers. However, once these belongings are removed and the locker no longer in use, the lock's combination is often forgotten. This leads to the lock being set aside, and when the need for a lock resurfaces, a new one is typically purchased. Our invention, the Combination Lock Solver, addresses this issue.

The Combination Lock Solver is a user-friendly device designed to solve forgotten combination locks. It can handle both attached and unattached locks with ease. To operate the device, users must first plug it into a wall outlet, allowing it to calibrate its unlocking mechanism. The dial is then set to the "Start" option. The user follows a series of steps: zeroing the dial, waiting for the device to identify the first combination number, zeroing the dial once more, and finally waiting approximately 12 minutes for the device to crack the lock's combination.

In cases where the user retains some knowledge of the first combination number, they can input this into the device. After zeroing the dial again, the user waits for the device to solve the remaining part of the lock combination. During this 12-minute solving phase, the device uses an algorithm to generate a list of 100 potential solutions, based on the provided first combination number.

Fig. 1 The Combination Lock Solver.

II. Design Process: Analsys, Testing, and Evaluation

The list of functional requirements for the Combination Lock Solver design is detailed as follows:

1. The device can determine the first combination number of a given combination lock.

2. The device can solve a given combination lock within ten minutes.

3. The device has a dimension requirement of less than 0.3 meters in length, 0.3 meters in width, and 0.3 meters in height.

4. The device weighs less than seven kilograms.

To determine the first combination number of the lock four solutions were proposed. When determining the first combination number, the pressure must be applied upward on the shackle as if it were being pulled open while the dial is slowly turned clockwise. As the dial turns, there will be multiple points along the number range where the dial meets resistance. To reduce the number of resistance points to just one, the pressure on the shackle must be released gradually until only one resistance point remains. If this resistance point occurs between two integers, the number is rounded up to the nearest whole number, and then the number five is added to it. This is how the first combination number is determined.

The first proposed solution included the use of a current sensor to measure the amount of current supplied through the voltage line of the stepper motor attached to the dial, to detect spikes in the stepper motor current. The idea behind this method is that when the dial experiences resistance, a back electromotive force is induced in the coil of the motor, causing the current output of the motor to rise rapidly. A test was conducted with the current sensor attached in series to the voltage line of the stepper motor where the motor was commanded to turn at a constant rate while the experimenter held the shaft in place to mimic the resistance observed by the dial. The current was graphed over time and was found to be so incredibly noisy that no conclusion could be drawn from the data to detect at which instance the resistance occurred at and ultimately which number on the dial it occurred at.

Fig. 2 ACS712 Current Sensor.

The second proposed solution to determine the point of resistance on the dial was to use a pressure sensor that was to be fixed in between two specially designed gears that spun together. The idea behind the design was that as the dial spins, so do the two gears and a pressure reading is recorded as the two gears push on each other. When the dial experienced resistance, the gear on the dial would exert a greater than normal force on the second gear to which the pressure sensor was to be placed in-between. The point of the pressure increase would indicate resistance in the dial which would be recorded as the resistance point. This design was not tested since it was determined to be too complicated to design and the group speculated that its design could likely cause the volumetric functional requirement to be violated.

The third proposed solution for resistance point determination was to include a sound sensor in the design which would listen for the clicks the lock made as it reached its resistance point. This method was tested with a benchtop design during with the lock was held in place, spun with the stepper motor, while a sound sensor was placed underneath the lock. Ultimately, the clicking sound made by the lock was not load enough for the sound sensor to detect so the design was ruled out. Another reason not to use the sound sensor was the possibility of it detecting sounds from the environment which could potentially affect the device’s ability to detect the correct resistance point of the third combination.

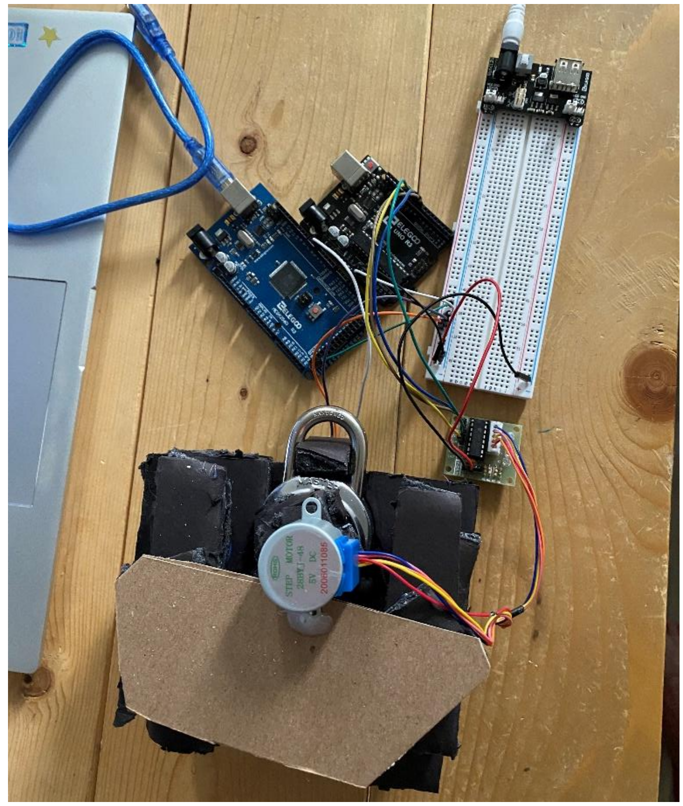

Fig.3 The benchtop setup using the sound sensor for resistance detection.

The fourth and final solution proposed to detect the resistance point of the lock included the use of a 28STH32 NEMA-11 dual-shafted stepper motor with a HKT22 optical encoder. One shaft of the stepper motor was fixed to the dial of the lock via a 3D printed PLA connection adapter piece while the other end of the shaft was connected to the optical encoder via a set screw. This ensured that as the dial shaft turned, so did the shaft of the stepper motor as well as the optical encoder. To detect the resistance point, the stepper motor was commanded to turn the dial five steps at a time which is equivalent to one number on the dial and 30 steps on the optical encoder. As the shaft of the stepper motor spun, if the optical encoder turned less than 30 steps, this would indicate that the shaft didn’t turn the commanded amount, indicating that the dial experienced resistance. This number would then be recorded as the resistance point.

Once the device determined the first combination number either via user input or the resistance detection function, the list of 100 possible combinations is generated. The algorithm for this determination is described as follows:

1. Divide the first combination number by four and save the remainder (either zero, one, two, or three).

2. If the remainder is zero or one, add two to it and record it. If the remainder is two or three, subtract two from it and record it. This number is used to determine a list of ten possible second combination numbers.

3. To determine the list of possible second combination numbers, divide the numbers zero through 39 by four and take note of which of those numbers has a remainder equal to the number saved in Step 2. These ten numbers will be the possible second combination numbers.

4. To determine the list of possible third combination numbers, divide the numbers zero through 39 by four and take note of which of those numbers has a remainder equal to the number saved in Step 1. These ten numbers will be the possible third combination numbers.

There is one possibility for the first number, ten possibilities for the second number, and ten possibilities for the third number. Taking the product of these three numbers gives a total of 100 possible combinations which the device iterates through and tests.

A stepper motor was chosen for this function since it allows for precise control of the position of its shaft compared to a normal DC motor. For example, a stepper motor can be commanded to move a precise amount and will stop moving once commanded to and will hold its position due to its coils being energized. A DC motor, however, would not stop once commanded to due to its rotational inertia.

Another key design consideration for the resistance point detection was the arm which lifted the shackle. Within the arm was a spring which would compress as force was applied to it. The spring was included in the arm design to add variable force to the shackle via the arm by commanding the arm to move up a variable distance. This variable distance compressed the spring which put a force on the shackle which is governed by this equation:

$$F = k\Delta x$$

Equation 1 describes the relationship between the force exerted on the shackle by the arm F, where k is the spring constant (N/m) and Δx is the compression of the spring.

The spring chosen for this design had a spring constant of one newton per meter.

Another key function of the device was to apply force to the shackle arm and to verify whether the lock was unlocked or not. To determine the amount of force required to pull open the shackle of a solved combination lock, a test was set up. During the test, the combination connected to a fish scale and then manually solved. One experimenter held the fish scale in place on a table while another experimenter pulled the lock until the shackle opened. The maximum force measured on the fish scale was recorded. This test led to the selection of a 20 kg-cm analog feedback servo motor to open the shackle since it met the force requirement to open the shackle.

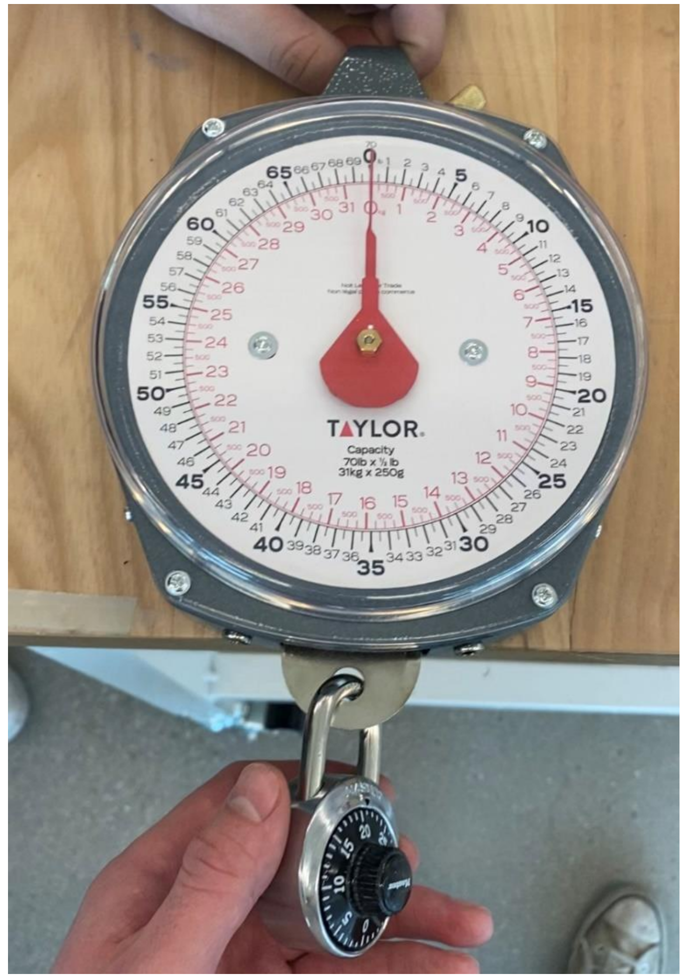

Fig. 4 Here is the test to determine the force required to open the shackle using the fish scale.

To determine whether the device unlocked the lock or not, there would need to be some type of feedback information. Three solutions were proposed for this problem.

The first solution was to use a system of an arm that opened the lock which also functioned as the rack of a rack and pinion gear system. The pinon part of the gear was essentially a spur gear fixed to the horn of the servo motor. Attached to the first spur gear would be a second spur gear whose axis was to be aligned with a potentiometer to measure the rotation of the gear. If the potentiometer was turned passed a set threshold value, this would indicate arm/rack gear moved high enough to unlock the shackle. If this threshold value was not met after a certain time, the device would determine that the shackle was not opened. This solution was originally tested in an early prototype; however, it resulted in the device being too large which led to it not being chosen.

The second solution was to implement a push button into the device that would only be pressed if the shackle was lifted into a locked position. This design was not chosen due to the lack of time to redesign the unlocking mechanism and the added complexity of the push button to the circuit of the device.

The third solution was to use an analog feedback servo to report the position of the servo arm in a similar rack and pinion style system with only one spur gear attached to the servo motor. As the servo arm was commanded to a certain position, the actual position of the arm was read. If this sensor value passed a certain threshold within a certain set time, then the lock was determined to be opened. If the threshold was not reached in the set time, it was determined to be locked.

III. Discussion

The product can solve a given combination lock in about 12 minutes, however, there are some limitations. While conducting many tests of the design, there were some obvious flaws that needed to be addressed and still do. The first is that the device only occasionally detects the first combination number. Sometimes, the device seems to apply too much pressure to the shackle, causing the dial to have multiple resistance points as it turns. At other times, the device seems to apply too little pressure to the shackle, causing the dial to have no resistance points. This is the result of not being able to move the rack arm by very fine amounts. This problem could potentially be solved by reducing the gear ratio between the rack and pinion gears so that the rack arm moves less distance per degree turn of the servo, allowing for fine adjustments. One issue encountered with this solution was that as the gear ratio was reduced, the higher the likelihood of slipping gears was observed. Between one iteration and the next after reducing this gear ratio, it was observed that the servo moved around a bunch and needed to be secured down to the bottom panel of the device with screws.

Another issue that was encountered with the device was in the 100-combination iteration solving function of the device where the stepper motor turned the dial. Often the dial was found to spin to the incorrect number which was measured with the optical encoder. This was attempted to be corrected in software where the stepper motor was to be commanded to move to the offset error determined by the optical encoder after each successive turn. This solution improved the solving function; however, it could be further improved with more software changes by comparing the expected and actual position of the dial at more frequent discrete timesteps for increased accuracy due to more feedback control. In addition to this, one potential reason the dial experienced resistance was when the shackle position was left in a “pulled-up” position after each successive try to unlock. This resulted in the arm design being changed to include a lever to pull the shackle back down to reduce the possibility of dial resistance while it was being turned. With improved feedback control in the solving function, the stepper motor speed could be increased and would result in the solving time dropping below ten minutes.

A third issue encountered was with the determination of the shackle state being locked or unlocked. The analog servo did not work as well as expected after it started reporting values that indicated that it unlocked the shackle when in fact it didn’t and vice versa. A better implementation would have been to include the push-button mentioned previously.

The last major issue with the device was the physical enclosure of the device. There was ultimately not enough room for all the electronics to fit nicely. In addition to that, the final 3D printed design for the bottom panel did not secure the servo well enough to the device to prevent any slipping of the rack and pinion gears. A work around solution to this was made by adding additional scrap pieces of PLA to the device to fix the servo down to the bottom panel using various wood and construction screws. More support should be added to the final design of the bottom panel to better secure the servo to the device.

IV. In The Marketplace

The Combination Lock Solver is a product most likely to be found in a convenience store or a gym that uses lockers for personal storage. Customers would enter these places and pay a small fee to use the device to solve their old combination locks. Individuals could also purchase the device; however, it is not recommended for the sake of misuse of opening the lock on a locker belonging to another individual. Similar devices that could evolve from this device are those that could be used to crack combination locks on safes or other similar locks. Competitors include any other type of combination lock solver.

V. Conclusion

Overall, the device worked on occasion and ultimately still needs work before becoming a viable product for the market. The first functional requirement is met inconsistently since sometimes the device could determine the first combination lock, and sometimes it failed to do so. The second functional requirement was also not met since it takes about 12 minutes to iterate through all 100 possible combinations. Next steps include redesigning the bottom panel to ensure the servo does not move and the gears always remain in contact. A push button needs to be added to the device to detect whether the shackle has been opened or not. The resistance detection needs to be improved by decreasing the gear ratio of the rack and pinion gear to allow for fine tune adjustments for shackle pressure as well as software changes to the method to allow for more consistency including multiple successive detections of the resistance point for increased confidence in the detection. And finally, the solving function needs software changes to increase the amount of position tracking of the dial using the rotary encoder and stepper motor.