Design and Development of a Mechanism for *Rubik's' Cube* Solution Aiming Robotics' Teaching

Plácido A. Souza Neto[1], Camila J. Ribeiro Teixeira, Helora D. Cruz Monteiro, Samara R. Silva, João M. Vilas Boas[1], Sarah T. de Lima Sá[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN) – Natal – RN – Brasil

{ placido.neto, joao.vilasboas, sarah.sa}@ifrn.edu.br

Abstract. *This paper proposes a low cost microcontrolled system to solve rubik's cube. This prototype is intended to be used in classrooms for improvement in the teaching-learning process, and consequently in the performance in disciplines of logic and algorithms.*

1. Introduction

In the last decade, with the increasing of technological advance, there is a need for a convergence between school *curriculum* and technological innovations. It happens because the students, born in the 21st century, find themselves immersed in an extremely technical world in all society domains. Additionally, the learning/teaching process of logic and basic programming in Brazil's institutions is still a huge challenge, either in technical and higher levels. This worrying truth, allied with the perception of the need to use innovation technologies in the school curriculum, creates an interesting scenario for the application of new approaches and pedagogical practices, especially regarding the use of robotics educational as a form of improvement in the teaching/learning process.

According to Soares et al. (2015), the school is the environment for the development of technology which contributes to the integral student formation, fostering critical thinking, stimulating the manipulation of elements in practical activities, instigating creativity and awakening the students the desire to produce knowledge.

Vilas Boas et al. (2016) affirm the learning difficulty mentioned above has been observed in the Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Natal Central Campus (IFRN / CNAT) in a very expressive diagnosis in the discipline of Fundamentals of Logic, considering courses of the technical level. From the data analyzed by them, they verified that the students failed percentage average in this discipline between 2012 and 2016 was 24%.

In this context, the objective of this work is the creation of a low cost articulated robotic mechanism, able to solve the rubik's cube problem through the implementation of algorithms. With this, it is expected to overcome the scenario presented with an increase of interest and performance in disciplines logic and basic programming using this prototype in the classroom. There is also the possibility of collective construction of the solution by groups of students involved.

2. The Rubik’s Cube

The rubik’s cube, one of the most popular puzzles in the world, came out in 1974. It is considered one of the most successful puzzles in history, mainly because it has more than 350 million units sold worldwide, making his inventor, the Hungarian Erno Rubik, become one of the richest men in his country.

Erno Rubik was born in the city of Budapest. While he was a university professor of interior design, he decided to create an object that would help him to perfectly illustrate the concept of the third dimension. From the Erno’s research by the expression and the increasing intensity of some thoughts, the interest for the cube was born.

Still in the 70's, the teacher was awarded for his invention, a fact that further stimulated people's interest in the toy. With the increase in demand, the puzzle began to be distributed throughout America, with a completely redesigned version, with easier faces to handle and more vivid colors, so it beings the called the *Rubik's cube*.

It is estimated that in the world, one in seven people have already played with the cube, and nowadays, various toy enthusiasts continue to produce cube-based puzzles such as Megaminx (dodecahedron), Pyraminx (tetrahedron) and 2x2x2 cube (a variation of the traditional cube 3x3x3), see figure 1.

(a) (b) (c)

Figure 1. Puzzles: (a) Megaminx; (b) Pyraminx; (c) Cube 2x2x2

2.1. God’s number

God's number is a reference to the minimum number of moves to solve any configuration of the rubiks cube. Solving it at random can be quite time consuming work, as there are exactly 43.252.003.274.489.856.000 possible combinations.

According to the site Cube 20, in July 1981, the mathematician Morwen Thistlethwaite constructed a complex algorithm and very difficult to be memorized. Unlike traditional algorithms such as layering, this method does not put piece by piece in its correct position, but works so that all pieces are placed in their correct positions at once. From its algorithm, Thistlethwaite has been able to prove that a total of 52 moves are needed to solve the puzzle.

In December 1990, the number was reduced to 42 movements. This fact was proved by Hans Kloosterman. Two years later, Michael Reid managed to slow down the minimum number of moves, this time to 39. In the same year, Dik Winter reduced that number to 37.

A few years later, in 1995, Michael Reid showed that God's number was 29. It took eleven years for that number to drop to 27, a fact proven by Silviu Radu. It is interesting to note that from 1981 to 2006 this number decreased by 25 movements, practically half the value that was initially proven. Still according to Cube 20, in May 2007, Dan Kunkle and Gene Cooperman showed that in fact the minimum quantity was 26 moves. The following year, Tomas Rokicki reached number 25. After, Rokicki and John Welborn reached 23 and then in 22 moves. In 2010, Rokicki, Herbert Kociemba, Morley Davidson and John Dethridge concluded that God's number is exactly 20. So far no one has been able to prove less than that.

2.2. Solution methods

As previously stated, the rubiks cube has more than 40 quintiles of possible combinations, and since the 1980s this three-dimensional puzzle has challenged many people to find a solution. There are a myriad of ways to solve the cube. Some people solve without a techniques aid, however, this is a laborious and tiring practice and much more difficult than just following the steps that are given in the methods.

Once learning an algorithm, it is essential to repeat the movements and understand the purpose of each step until it becomes natural and intuitive. The methods range from the simplest to the most complex, and it influences the time that will be spent in the solution. Some sequences are derived from other methods or even are combinations of several of them.

There are several tutorials on Brazilian websites, the most popular is the Speed Cube[1], developed by Renan Cerpe, and the site belongs to one of the WCA (World Cube Association) delegates, Rafael Cinoto. The French site Francocube [2]is one of the most outstanding worldwide, for being complete and presenting tutorials for different types of different puzzles. Following we will be expose some of the most used methods to solve the rubiks cube (McNaughton, 1990).

a) *Layer method*

The layer method is one of the most used by those who are starting to learn how to solve the puzzle. It is a very simple technique that does not present great complexity. It is divided into seven steps that aims to solve the cube layer by layer. This method was proposed in the 80's and performs an average of 100 movements. The algorithm will be explained in details later.

b) *Intermediate Method*

The intermediate method is an extension of the traditional layer method. Its difference is that it has some more advanced algorithms to solve the last two layers, this reduces the number of movements generated and the time spent to solve the cube, a total of 30 to 40 seconds less than the previous method.

---

1 https://www.rubiks.com/speed-cubing

2 https://www.francocube.com/

c) Fridrich method

The Fridrich method, developed by Jessica Fridrich, is one of the most used by rubiks cube competitors worldwide. Nowadays it is considered one of the fastest method to solve the cube. However, it is also one of the most difficult because it uses several algorithms. Its variation to the intermediate method is that it has even more algorithms that serve as shortcuts to the traditional algorithm, which further reduces the time and the number of movement.

d) Petrus method

The Lars Petrus method is also an advanced method. It was developed in the 80's and it is based on the resolution by block construction, where the first two layers are solved in a totally intuitive way. It is the second most popular advanced method, after Fridrich. This method is also partially used in other methods. In some situations, this method may generate fewer moves than Fridrich, but sometimes, especially for beginners, it can be difficult to due to the large number of intuitive movements.

3. Mechanism description

Following, we will briefly describe the hardware and software solution developed for the proposed system.

3.1. *Hardware*

The hardware structure of the robot consists of parts made by a 3D printer in ABS material, a base for the hub, a horizontal movement arm and an adjustment claw on the cube. These materials are arranged in an area measuring 49.5 cm in length and 15 cm in width. The cube used for the design is the Shengshou Stickerless, Rainbow Stickerless 3x3 model. The operation of the robot consists of the mechanical arm pushing the cube to change to the next face, and to exchange each face the arm holds the hub while the base rotates 90, 180 or 360 degrees. Then the adjusting jaw flushes the cube.

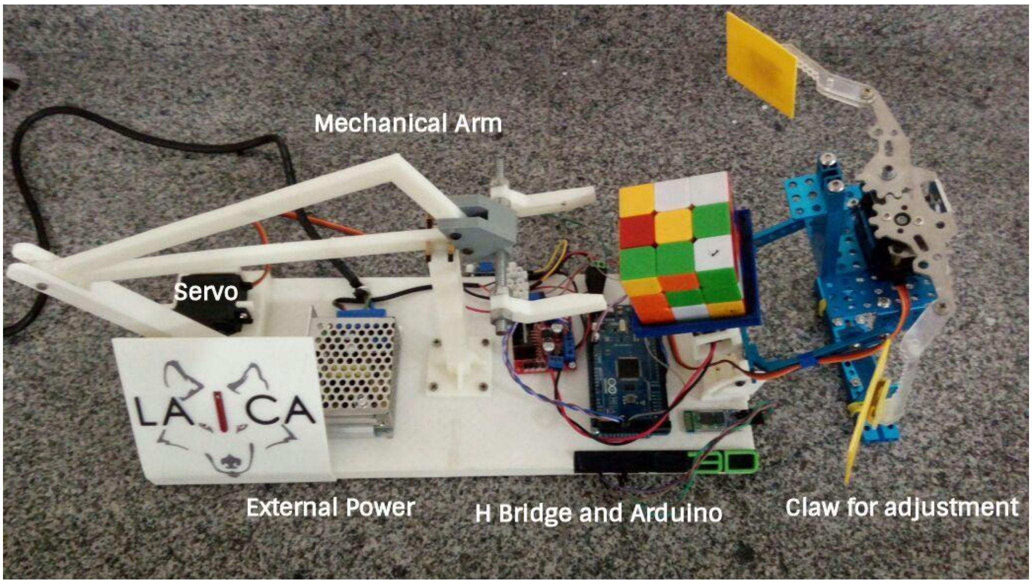
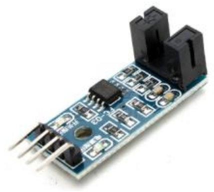


Figure 2. Articulated mechanism with servo motors and control plates.

The horizontal movement mechanical arm was built with the 3D printer and has 29.5 cm, its main functions are to push the cube so that it changes face and hold it when it is necessary to change the pieces of place. The mechanical arm moves through a servo motor TowerPro, MG995 model. We also use the same type of servo to move the base of the cube. The base was built into the 3D printer and has 6.7 cm width and 6.7 cm height and is responsible for rotating the hub at positions 90, 180 and 360 degrees. To perform these movements, coupled to the base, there is an optical encoder of 5 mm and a rotating disk that sends data to the encoder.

To give stability to the robot base, there is a 23.5 cm height metal frame made with pieces of the educational robotic platform Makeblock. At the top of this structure there is a TowerPro servo motor coupled to a modified mechanical Claw Robot Mechanical Claw H3. The claw is responsible for adjusting the sides of the cube when its parts change places. All servo motors are connected to a source Beehive Model S-30-12 of 5 amps and 12 volts. The power supply also powers a H-bridge L298N, its function is basically to control the speed and direction of the motors. The H bridge is connected to a voltage regulator LM2596 model, responsible for regulating the reactive power division between the devices. All these equipment are connected to an Arduino Mega 2560.

(a)



(b)

(c)

(d)

Figure 3. Some parts of the proposed system: (a) TowerPro servomotor MG995 model, (b) encoder optic, (c) Bridge H L298N model, (d) Arduino Mega 2560 model.

3.2. *Software*

The basic method, also known as layer method, is simple and widely used by those interested in learning how to solve the rubiks cube. As its name suggests, this solution consists of solving the cube layer by layer, using only seven steps.

- Step 1: make the cross in the first face;
- Step 2: position the corner pieces of the first layer;
- Step 3: position the edge pieces of the second layer;
- Step 4: make the cross on the top face;
- Step 5: review the top corner pieces;

➢ Step 6: permute the corner pieces;

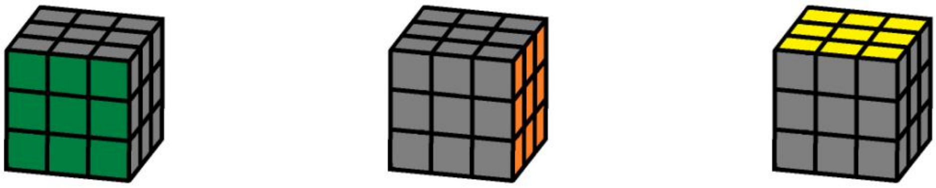
➢ Step 7: permute the edge pieces.

3.2.1. Notations

Before describing the algorithms that make up the layer method, it is essential to define some concepts and nomenclatures. Singmaster (1981), published one of the first analyses of the Magic Cube. He introduced the following notation:

a) Face

The WCA, which regulates the rubiks cube competitions in the world, establishes an official notation for each of the six faces of the traditional 3x3 cube, where each one is represented by a letter: F (front), L (left), R (right), B (back), U (top) and D (bottom).



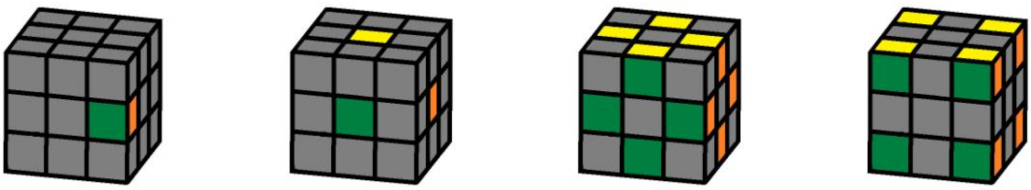
(a) (b) (c)

Figure 4. Faces: (a) Face F; (b) Face R; (c) Face U.

b) Piece

Each piece is formed by the colors of the faces that compose it. An example is the middle piece that belongs to faces F and R, which can be called FR or RF (see figure 6a).

The cube consists of three types of pieces. One is the center piece, which is fixed and indicates the color of a given face. The others are the edge pieces, which have two colors, and the corner, which is composed of three colors. It is important to note that the parts can only be positioned in place of parts of the same type.



(a) (b) (c) (d)

Figure 5. (a) pieces FR; (b) Centers; (c) edges; (d) corners.

c) Movements

Each face produces three types of basic movements: the 90° clockwise, 180° and 90° counterclockwise rotation. The representation of these turns is done using the following symbology: if it is a rotation of 90º clockwise we write only the letter that corresponds to the face; For the rotation of 180º we put the letter plus the number 2; And for the rotation of 90° counterclockwise we write the letter plus' (apostrophe). For example, the face U produces the following rotations, *e.g.*, U, U2 and U '.

(a) (b) (c)

Figure 6. Turns: (a) U; (b) U2; (c) U'.

d) Permutation and Piece Orientation

The concept of permutation is related to the positioning of the pieces in a specific state that is the cube. It is nothing more, but exchange one piece for another. Orientation occurs when the piece is in its correct position, but the cube is still not solved

3.2.2 Representation

The concept of matrices was used to represent the faces of the cube. Considering that the rubiks cube consists of six faces, six arrays of integers were created, each with three rows and three columns.

Each color of the cube was symbolized by an integer and each face was represented by a color, as shown in Table 1.

Table 1. Rubiks cube face representation

Face	Color	Number
D	White	1
U	Yellow	2
F	Green	3
R	Orange	4
B	Blue	5
L	Red	6

3.2.3. Solution

The implementation of the layer algorithm was made in C language. The cube and its faces were represented using the struct concept. A variable was also created responsible for storing the movements generated in the solution.

A method was implemented for each type of movement. These methods take one face and spin the edge and corner pieces. Below is the code snippet for 90 ° clockwise rotation.

Three methods were created for each face, each representing the possible turns that the faces can perform. For example, for face F the following methods were created: frontClock (clockwise 90 °, or F), frontClock180 (180 ° rotation, or F2) and frontIClock (clockwise rotation 90 °, or F ').

```
void frontClock(int left_f[][3], int front_f[][3], int right_f[][3], int top_f[][3], int bottom_f[][3], int back_f[][3]){


   turnClock(front_f);
   getBottom(top_f);
   getLeftI(right_f);
   getTop(bottom_f);
   getRightI(left_f);
   setBottom(top_f, rightI_1);
   setLeft(right_f, bottom_1);
   setTop(bottom_f, leftI_1);
   setRight(left_f, top_1);
   strcat(historyMoviments, "F\n");


}
```

Figure 7. Code method for turn Front clockwise.

Each method receives (figure 7) the cube faces as a parameter and performs the expected rotation. At the end of each method the movement performed is added to the variable responsible for storing the movement history.

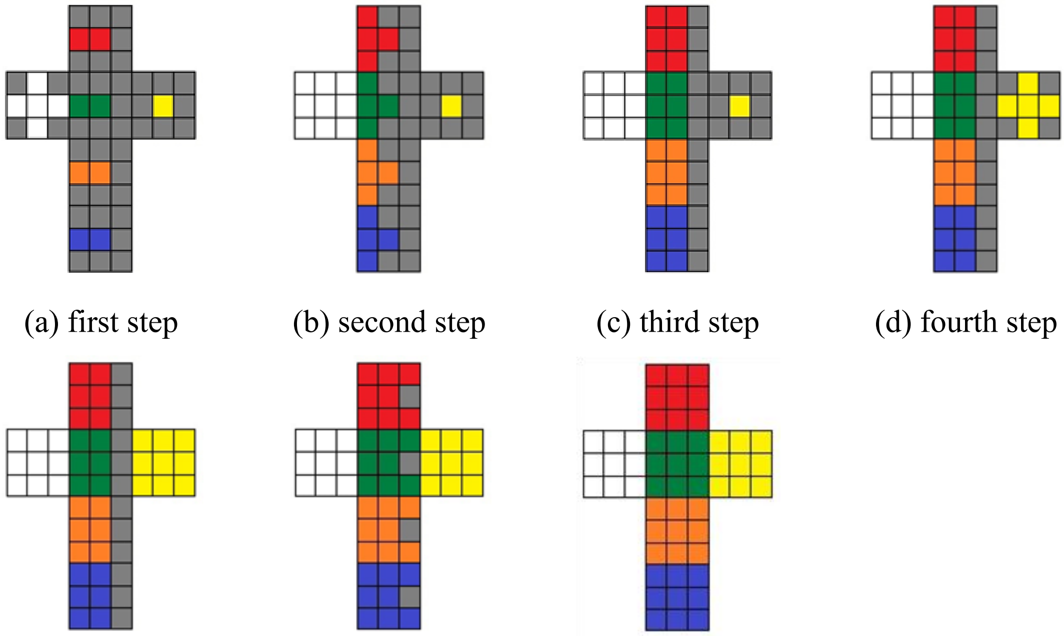
```
void firstStep(int left_f[][3], int front_f[][3], int right_f[][3], int top_f[][3], int bottom_f[][3], int back_f[][3]){


if (bottom_f[1][2] == 1 && right_f[2][1] == 3) {
      bottomIClock(left_f, front_f, right_f, top_f, bottom_f, back_f);
   } else if (bottom_f[2][1] == 1 && back_f[2][1] == 3) {
      bottomClock180(left_f, front_f, right_f, top_f, bottom_f, back_f);
   } else if (bottom_f[1][0] == 1 && left_f[2][1] == 3) {
      bottomClock(left_f, front_f, right_f, top_f, bottom_f, back_f);
   }
}
```

Figure 8. Code part to verify some conditions on first step in the Layer method.

In addition, we also implemented seven other methods that correspond to the steps of the layered algorithm. As the objective of this algorithm is to solve through the concept of layers, then the first face to be assembled was D.

The initial step is to make the cross, so the first method looks at all the possible positions that are the pieces of white medium. After this analysis, the methods necessary to change the positioning of the found part are called. The purpose is illustrated in the figure 7. Figure 8 presents a code example that corresponds to the first case that the magic cube can be configured.



(a) first step (b) second step (c) third step (d) fourth step

(e) fifth step (f) sixth step (g) seventh step

Figure 9. Rubik's cube solution steps.

To solve the second layer, the similar mapping is done on the matrix, this time looking for pieces of edge that correspond to the faces F, R, B and L (figure 9.b). With the first and second layer completed (figure 9.a and figure 9.b), only the last layer remains.

The last four steps of the algorithm are responsible for assembling the third layer (figure 9.c). First, it is necessary to cross the U face (figure 9.d and figure 9.e). This mapping is done by analyzing the parts that were not in their correct position and the parts that were in the correct but inverted position.

The fifth step (figure 9.e) finishes the face U of the cube completely, raising all its pieces of steel. In this step we do not worry about the parts of faces F, R, L and B of the last layer. For this step the cube can be in seven different configurations. In all cases the movements will be the same, changing only the amount of times it should be applied and the position of the cube.

In the sixth step (figure 9.f) we exchanged the four corner pieces from the last layer. The algorithm locates a side that has two corners of the same color and applies the required sequence of motions.

The seventh and last step (figure 9.g) makes the permutation of the middle pieces of the last layer. This algorithm searches for one side of the cube that is fully assembled, except for the face D, and verifies the direction the permutation should be applied.

All code produced to solve the rubiks cube problem is used as example to teach either Arduino and programming logic principles in technical courses. As well as we apply the results and experiences acquired in the projects to improve the way we recruit new student to be involved in robotic project.

4. Conclusions

The resolution of the magic cube through the mechanism proposed in this article was accomplished successfully, since the aforementioned robotic system, besides having a low cost, compared to other architectures present in the market, presented a satisfactory yield, obtaining solutions in less than 13 minutes.

Through the preliminary results obtained in this work, it can be observed that the use of the articulated mechanism to solve the problematic of the magic cube has shown to be promising and should be applied in the classroom for future verification of its effectiveness regarding the improvement of teaching-learning process in disciplines of logic and programming.

In this way, in future works, this methodology will be applied in disciplines of the programming area of technical and superior courses of the IFRN. The group is also studying the development of a smartphone application that allows the user to quickly enter the values of the cube faces into the system from image analysis. This implementation aims to streamline the process and make the human-machine interface more user-friendly, which should greatly improve its application in the classroom.

5. References

CUBE20, God's number is 20. Disponível em: <http://www.cube20.org/>. Acesso em: 27 mai. 2017.

MCNAUGHTON, D. (November 1989 - February 1990). "The Rubik Cube: A three- stage approach to mastering it". Junior News (Al-Nisr, Dubai, UAE).

SINGMASTER, David (1981). Notes on Rubik's Magic Cube. Harmondsworth, Eng: Penguin Books. ISBN 0-907395-00-7.

JOHANSSON, Frans. Clique: como nascem as grandes ideias. São Paulo: Portfolio-penguin, 2013.

SOARES, A. A. A. F. *et al.* (2015) "A preparação para a olimpíada de robótica como projeto", In: 6º Workshop of Robotics in Education, Uberlândia/MG, 2015.

VILAS BOAS, J. M. *et al.* (2016) "The use of a maze solver robot to support the teaching and learning process", In: 6º Workshop of Robotics in Education, Recife/PE, 2016.

KORF, Richard E. 1997. Finding optimal solutions to Rubik's cube using pattern databases. In Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence (AAAI'97/IAAI'97). AAAI Press 700-705.

KORF, Richard E. 1982. A program that learns to solve rubik's cube. In Proceedings of the Second AAAI Conference on Artificial Intelligence (AAAI'82). AAAI Press 164-167.

GOVINDA K. and PRASANNAS. A generic image cryptography based on Rubik's cube T2 - 2015 International Conference on Soft-Computing and Networks Security (ICSNS).

IONESCU, V. M. and DIACONU, A. V., "Testing the performance of the improved Rubik's cube encryption algorithm on virtual systems," 2015 14th RoEduNet International Conference - Networking in Education and Research (RoEduNet NER), Craiova, 2015, pp. 121-126. doi: 10.1109/RoEduNet.2015.7311980