A Novel Technique for Controlling CNC Systems

Hussein Sarhan

Department of Mechatronics Engineering, Faculty of Engineering Technology, P. O. Box 15008, Amman, Jordan

E-mail: sarhan_hussein@yahoo.com; hussein_74707@hotmail.com

Abstract

This paper describes a new CNC machine control unit MCU for numerically controlled NC systems used for machining or non-machining applications of numerical control NC. The proposed MCU is a combination of computer PC with Matlab software and Arduino microcontroller. The Matlab is used as an image processing program to accomplish the boarder detection of the image and convert it into digital output (0 and 1) for each pixel by a number of functions. The pixels will bias an array for the image, which will be processed to instructions. These instructions will be sent to Arduino microcontroller by the Matlab through Arduino cable (serial port). The Arduino microcontroller will generate the proper pulse trains to control the operation of stepper motors in the processing equipment (machine tool). The designed MCU has been implemented and experimentally tested in three-axes milling machine. Performance analysis of accomplished tasks verifies that the suggested approach for NC-part programming can be developed and used for different machining and non-machining applications of NC systems.

Keywords: numerical control, NC-part programming, Matlab, Arduino microcontroller, milling machine

1. Introduction

The term numerical control NC is a widely accepted and commonly used term in the machine tool industry. NC can be defined as a form of programmable automation in which the mechanical actions of a machine tool or other equipment are controlled by a program containing coded alphanumerical data. Thus, NC is any machining process in which the operations are executed automatically in sequences as specified by the program that contains the information for the tool movements. The alphanumerical data represent relative positions between a workhead and a workpart as well as other instructions needed to operate the machine. The workhead is a cutting tool or other processing apparatus, and the workpart is the object being processed. NC which quickly became computer numerical control CNC has brought tremendous changes to the metalworking industry. New machine tools in CNC have enabled industry to consistently produce parts to accuracies undreamed of only a few years ago. The same part can be reproduced to the same degree of accuracy any number of times if the CNC program has been properly prepared and the computer properly programmed. The operating commands which control the machine tool are executed automatically with amazing speed, accuracy, efficiency and repeatability (M. Tolouei-Rad, G. Payeganeh 2006, M. Tolouei-Rad, G. Payeganeh 2006). In general, NC has two main applications: machine tool applications, such as drilling, milling, turning, and other metal working and nonmachine tool applications, such as assembly, drafting, and inspection. The common operating feature of NC in all of these applications is control of the workhead movement relative to the workpart. The essential features of numerically controlled machines have been established for many years. They comprise a controller, known as the machine control unit MCU, capable of reading and interpreting a stored program and using the instructions in this to control a machine via actuation devices (Joze Balic 2004, Primoz Krzic *et al.* 2009). This arrangement is shown in the following Figure 1. The instructions (NC part program) is usually done through direct numerical control DNC connection, with punched tape through the tape reader, or manually, through interface for manual data input. The manual data input is used for simple machining operations or for corrections of NC part program and for changing technological machining parameters. The program of instruction is communicated to the machine tool using a coding system based on binary numbers. This NC coding system is the low-level machine language that can be understood by the MCU. When higher level languages are used, such as APT, the statements in the program are converted to this basic code. In addition to number values, the NC coding system must also provide for alphabetical characters and other symbols. Eight binary digits are used to represent all of the characters required for NC part programming. There are two standard coding systems currently used in NC: the Electronic Industry Association EIA and the International Standards Organization ISO. The EIA system is known as EIA RS-244-B. The ISO code was originally developed as the American Standard Code for Information Interchange ASCII and has been adopted by ISO as its NC standard.
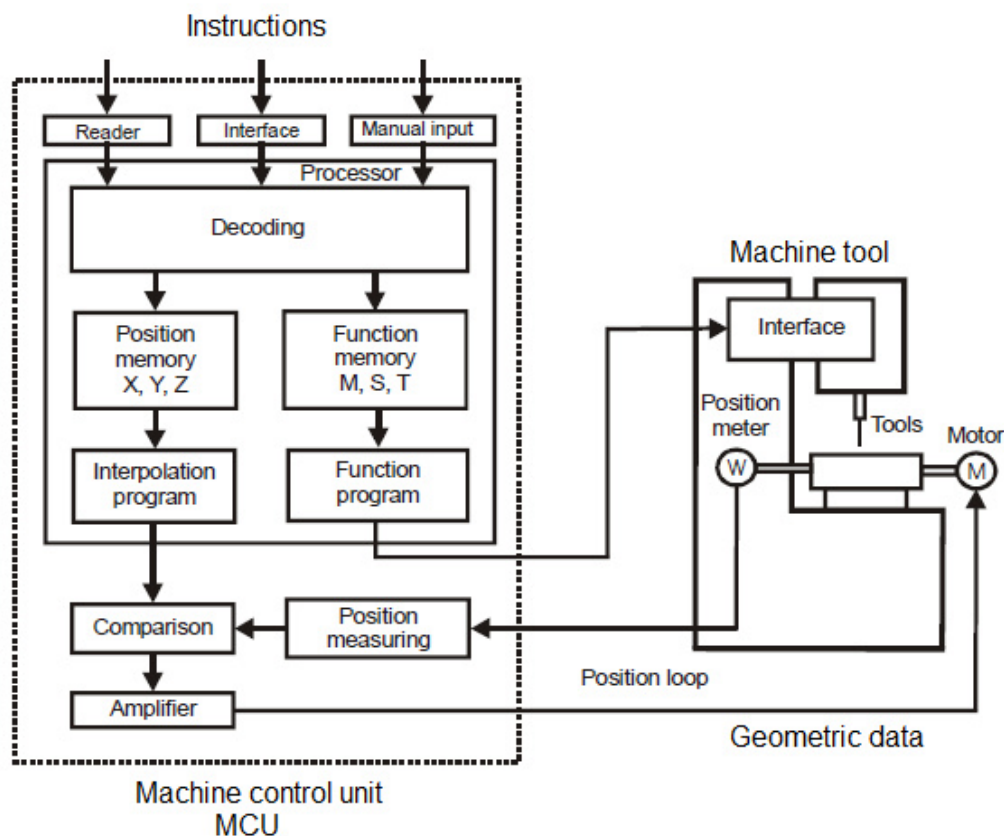
Figure 1. Structure of Conventional CNC Machine Tool

Every program for any workpiece to be machined must be put in a format that MCU can understand. The format used on any CNC machine is built in by the machine tool builder and based on the type of MCU on the machine. Word address is the most common programming format used for CNC programming systems. This format contains a large number of different codes (NC words) that transfers program information about the workpiece (geometry and machining conditions) to machine tool actuators. These codes, which conform to EIA standards, are in a logical sequence called a block of information. Each block should contain enough information to perform one machining operation (Primoz Krzic et al. 2009, J. Balic 2006, Chang. I., Deng. J., Chan. S. 2000). The most common codes used when programming CNC machine tools are G-codes (preparatory functions) and M-codes (miscellaneous functions). Other codes such as F, S, D, and T are used for machine functions, such as feed, speed, cutter diameter offset, tool number, etc. NC part programming may be performed manually (manual part programming) or by the aid of a computer (computer-aided part programming).

The ever-increasing use of CNC in industry has created a need for personnel who are knowledgeable about and capable of preparing the programs which guide the machine tools to produce parts to the required shape and accuracy. With this in mind, the purpose of this paper is to design and implement a novel technique for NC part programming, based on available Matlab and Arduino software to control the operation of CNC systems.

2. Software-Based NC-Part Programming Procedure

Figure 2 illustrates the procedure of proposed NC-part programming based on Matlab and Arduino software. The procedure consists of the following steps:

2.1. Image processing: Matlab software provides massive library and functions for image processing and manipulation which are used to get a black and white image with defined boarders of the geometry (shape), represented by digital numbers (1 or 0) for each pixel. Image processing composes the following actions:

1. Reading the image. While image is stored in the PC, it can be imported as a data to the Matlab by using the following code: Picture=imread('tray.jpg');

This image will be shown in the Matlab as an array of pixels, the colors will be represented as an array, and the elements of this array will be an integer number from 0-255. An example of image reading from graphics file is shown in Figure 3.
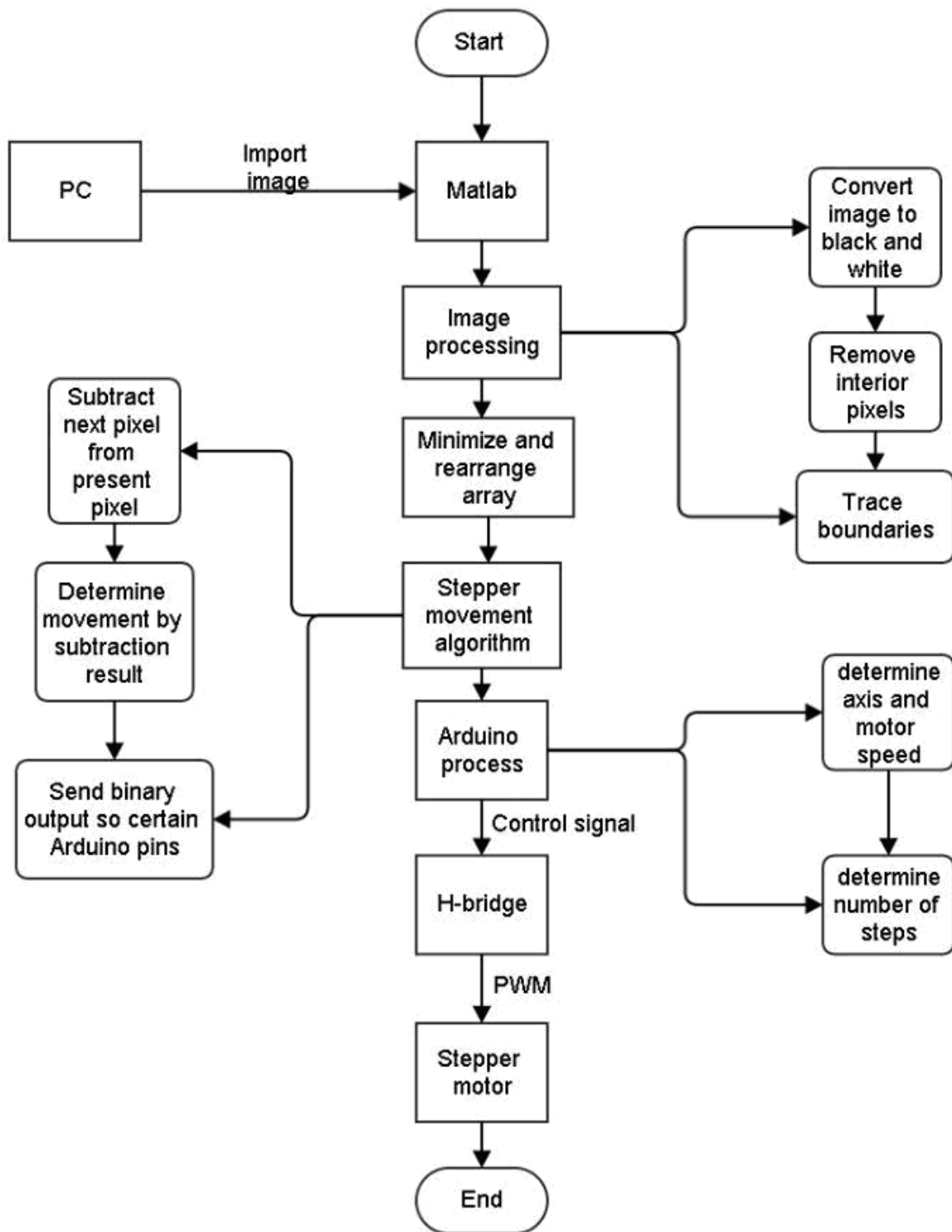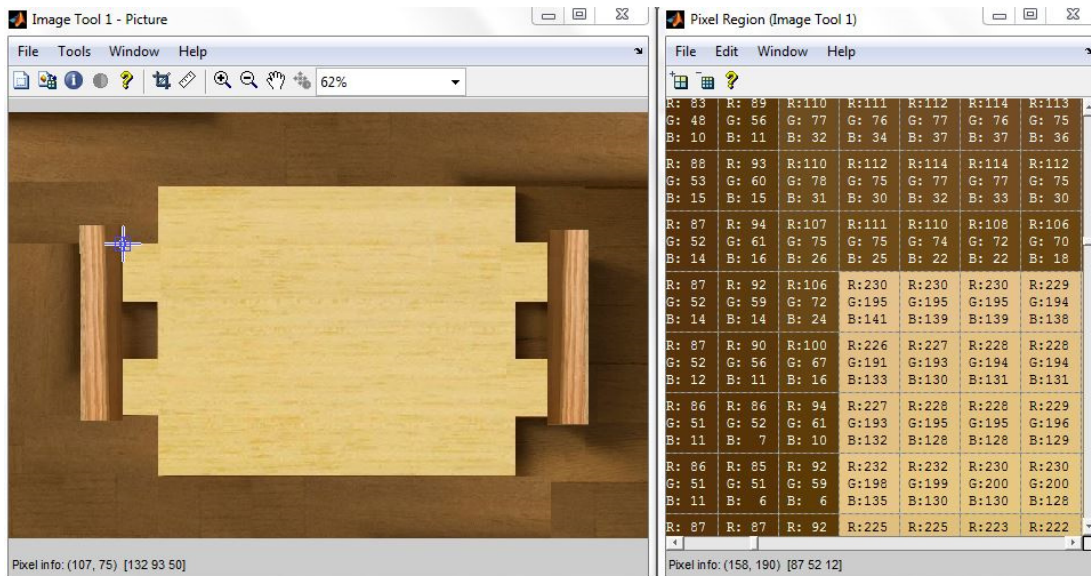
Figure 2. Flow Chart of Programming Procedure

Figure 3. Image Reading From Graphics File

2. Convert the image to black and white image, as shown in Figure 4, by using the code: Pic_bw = im2bw(Picture);
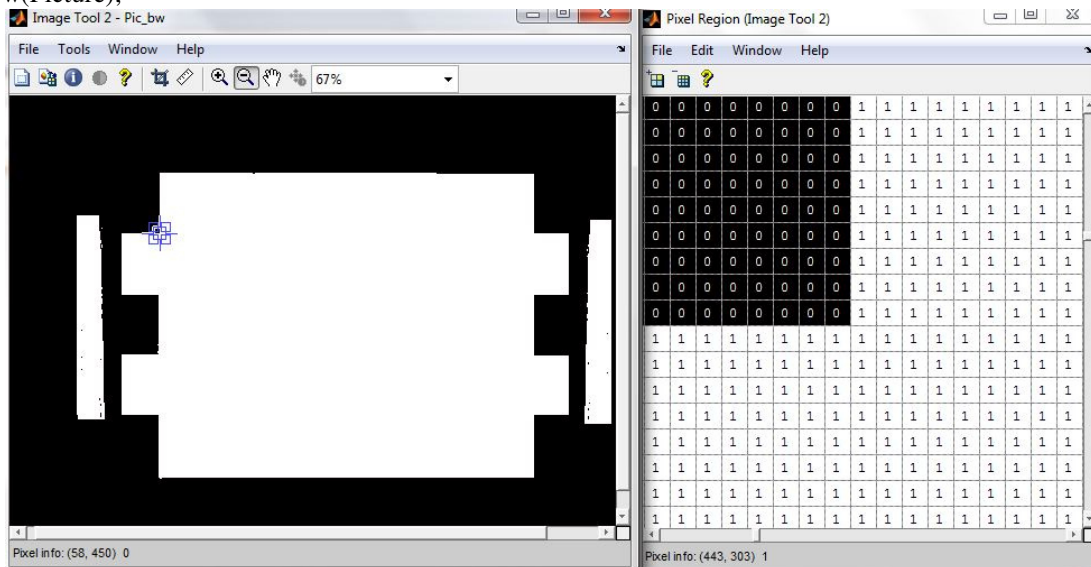

Figure 4. Conversion the Image to Black and White

3. Removing interior pixels, as shown in Figure 5, by using the code: Pic_CN=bwmorph(Pic_bw,'remove');
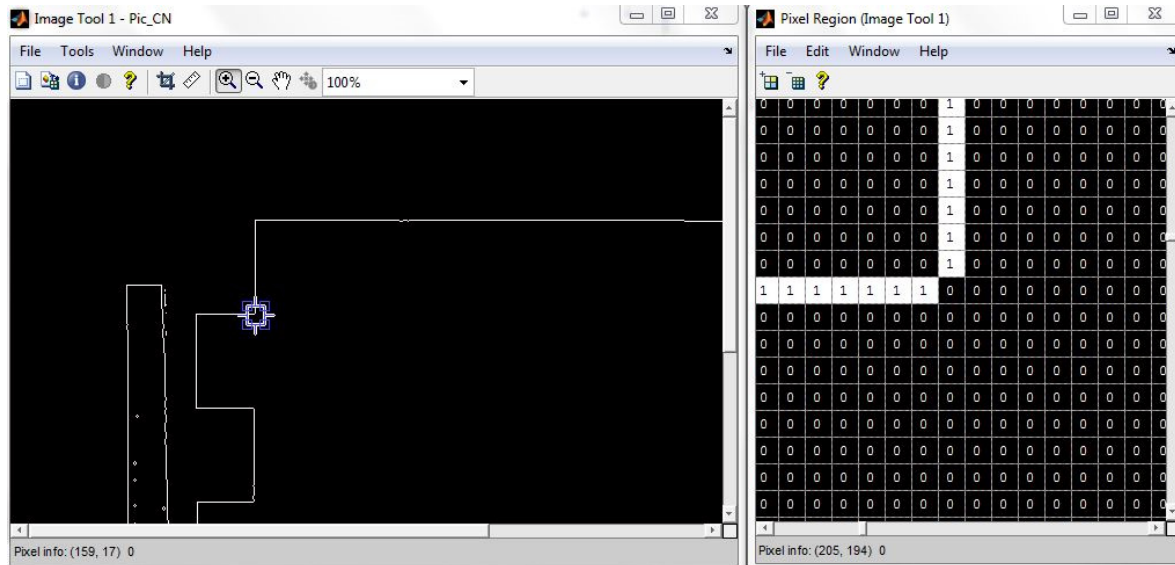
Figure 5. Removing Interior Pixels of a Binary Image

4. Tracing all of the outer boundaries for all shapes inside the image, as shown in Figure 6, by using the code:
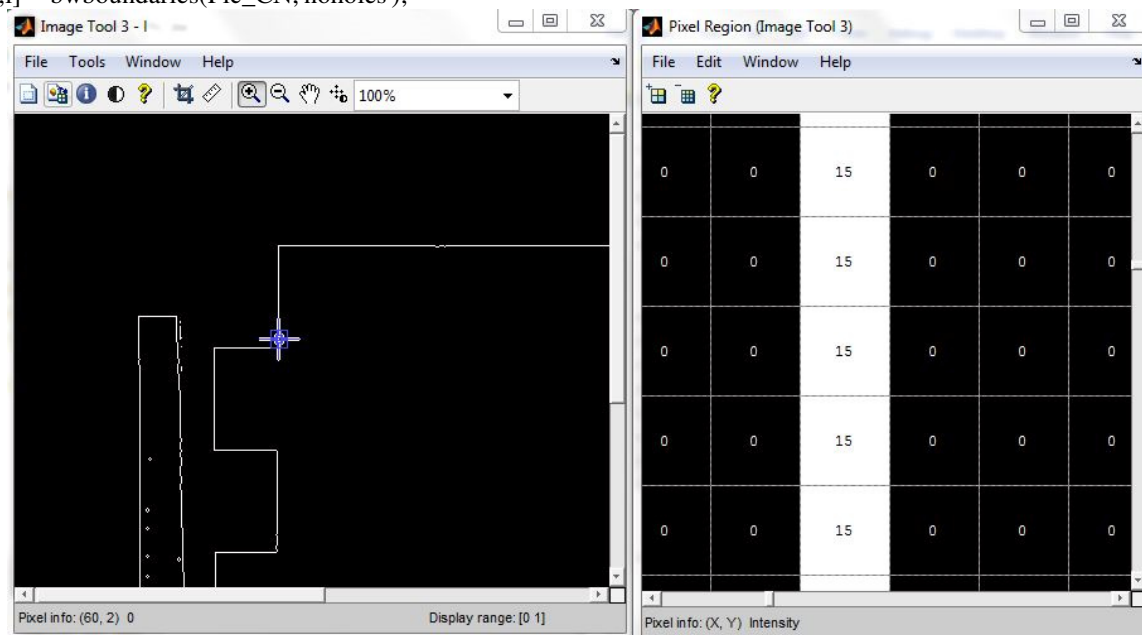[b,l] = bwboundaries(Pic_CN,'noholes');

Figure 6. Tracing the Outer Boundaries for all Shapes Inside the Image

5. Determining the boundaries as X, and Y coordinates, as shown in Table 1, by using the code:
[x,y]=find(l==15);

Table 1. Determining the Boundaries as X and Y Coordinates.

```
>> [x,y]

ans =

        18      44
        19      44
        20      44
        21      44
        22      44
        23      44
        24      44
        25      44
        26      44
        27      44
        28      44
        29      44
        30      44
```

6. Rearranging all boundaries and connecting them together, as shown in Figure 7, by using the code:
Final_boundary=cat(1,Pic_boundary_1,Pic_boundary_2);
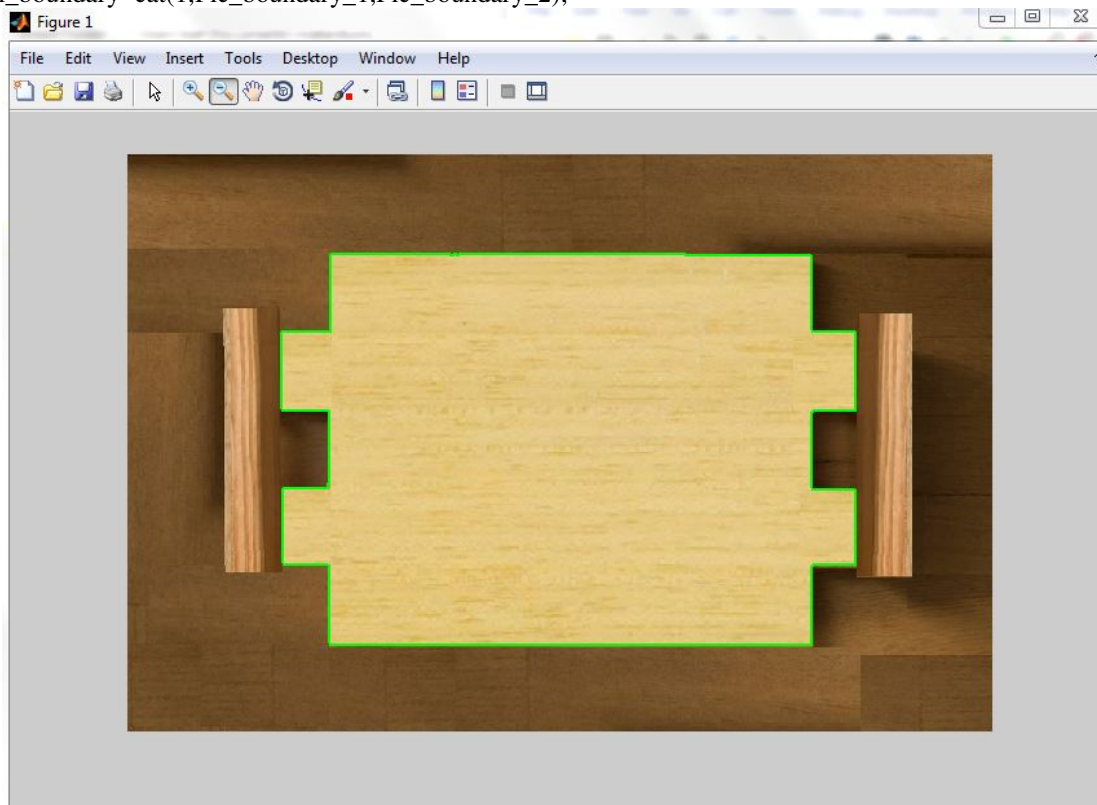

Figure 7. Rearranging and Connecting the Boundaries

7. Minimizing the array by taking every 5 pixels as 1 pixel, by using the following loop:
Last_row=final_boundary(:,1);
Last_column = final_boundary (:,2);
for i=1:5:length(last_row)
new_row(i)=row(i);
new_column (i)=last_column (i);
end
new_row=new_row(new_row~=0);%eliminate zeros in the array
new_column =new_column (new_column ~=0);

2.2 Communication between Matlab and Arduino using USB serial port. Transmission data from Matlab to Arduino is accomplished by the following actions:

1. Unify the baud rate (the number of bits transferred per second) between both Matlab and Arduino software.

2. Unify the start bit and stop bit between both Matlab and Arduino software.

3. Open serial communication for both Matlab and Arduino software using (Fopen) function for Matlab , and begin() for Arduino.

4. Make the communication serial ready, and start transfer bits.

5. Add the library to the Matlab folders.

6. Determine the used COM port.

7. Activate this library on Matlab by function :A=Ardino('Com4').

2.3 Stepper motors functions and movement algorithm. This step includes:

1. Activation of Arduino steppers library by using the function: **#include <Stepper.h>**

2. Defining the movement algorithm through direction, controlling and activation sub-algorithms. The direction sub-algorithm is based on subtraction operations. The next element (pixel) of the rows and columns is subtracted from the present one. This will produce one of the results, shown in Table 2.

Table 2. Stepper Motors Conditions According to Subtraction Results

Subtraction probability	Subtraction result		Condition
	Rows	Columns	
1	0	0	No change of movement
2	0	+	Move in straight line in the positive X-axis
3	0	-	Move in straight line in the negative X-axis
4	+	0	Move in straight line in the positive Y-axis
5	-	0	Move in straight line in the negative Y-axis
6	-	+	Move in diagonal line in the first quarter
7	-	-	Move in diagonal line in the second quarter
8	+	-	Move in diagonal line in the third quarter
9	+	+	Move in diagonal line in the fourth quarter

After determining the movement results, signals will be sent to certain pins. These signals will be translated into a stepper motor movement directions according to the Table 3.

Table 3. Arduino Pins Outputs According to Subtraction Probability Movement

Subtraction probability	Pin (20)	Pin (21)	Pin (22)	Pin (23)	Condition
1	0	0	0	0	Stop all motors
2	0	0	1	0	Move the X-axis motor backward
3	0	1	0	0	Move the X-axis motor forward
4	0	1	1	0	Move the Y-axis motor forward
5	1	0	0	0	Move the Y-axis motor backward
6	1	0	1	0	Move X-axis and Y-axis motors forward
7	1	1	0	0	Move X-axis and Y-axis motors backward
8	1	1	1	0	Move X-axis motor forward and Y-axis motor backward
9	0	0	0	1	Move Y-axis motor forward and X-axis motor backward
——	1	0	0	1	Move Z-axis motor forward
——	1	1	0	1	Move Z-axis motor backward
——	1	1	1	1	Turn the cutter ON

3. Activation sub-algorithm to determine the proper motor and the value of each axis (number of steps) by using the following commands:

```
x_1=digitalRead(20);
x_2=digitalRead(21);
x_3=digitalRead(22);
x_4=digitalRead(23);

if (x_1==0 && x_2==0 && x_3==1 && x_4==0)
 {
     stepperX.step(1);
 }

 if (x_1==0 && x_2==1 && x_3==0 && x_4==0)
 {
     stepperX.step(-1);
 }
```

These commands will be repeated until all pixels are drawn.

3. Experimental Setup

The proposed NC part programming approach has been implemented and experimentally tested for three-axes CNC milling machine. The block diagram of the used CNC system is shown in Figure 8. A picture of experimental setup is shown in Figure 9. The processing equipment can perform milling and drilling machining operation. Also, it can be used for automated drafting. The range of X and Y axes is 200mm.
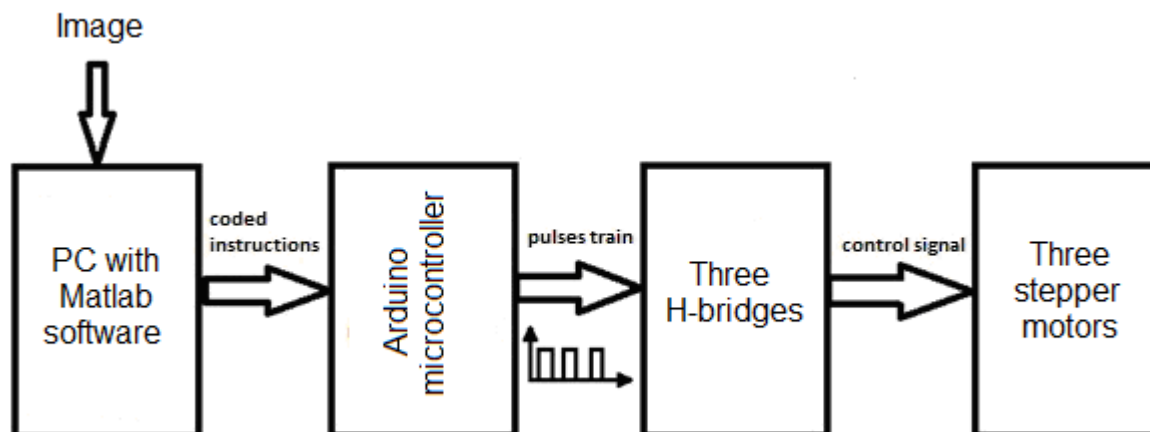
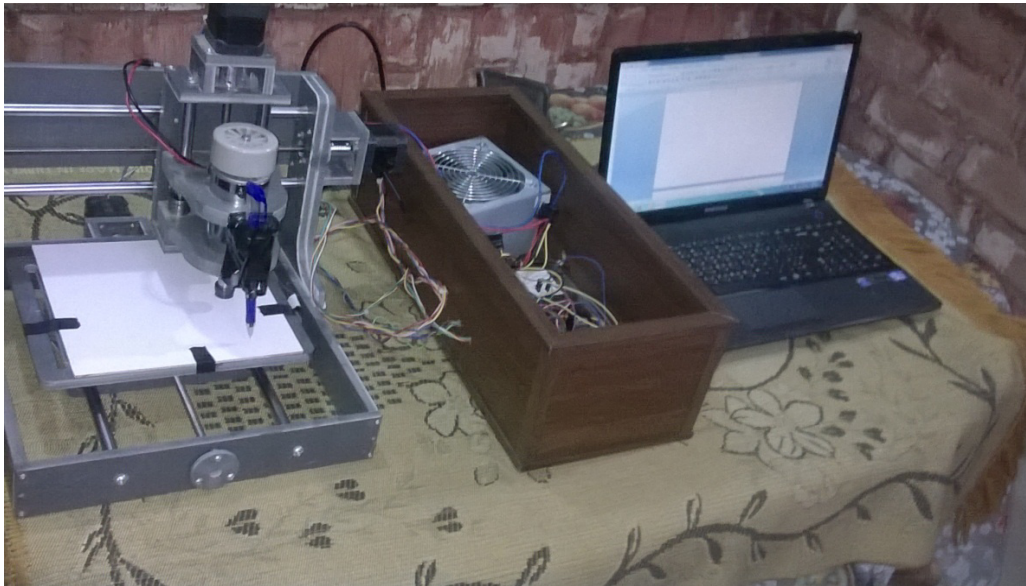Figure 8. Block Diagram of the Used CNC System

Figure 9. Picture of Experimental Setup

The system consists of the following main components:
1. Computer for processing and filtration on the required image by using Matlab programs and special functions.
2. Arduino Mega ATmega 1280 microcontroller, as shown in Figure 10.
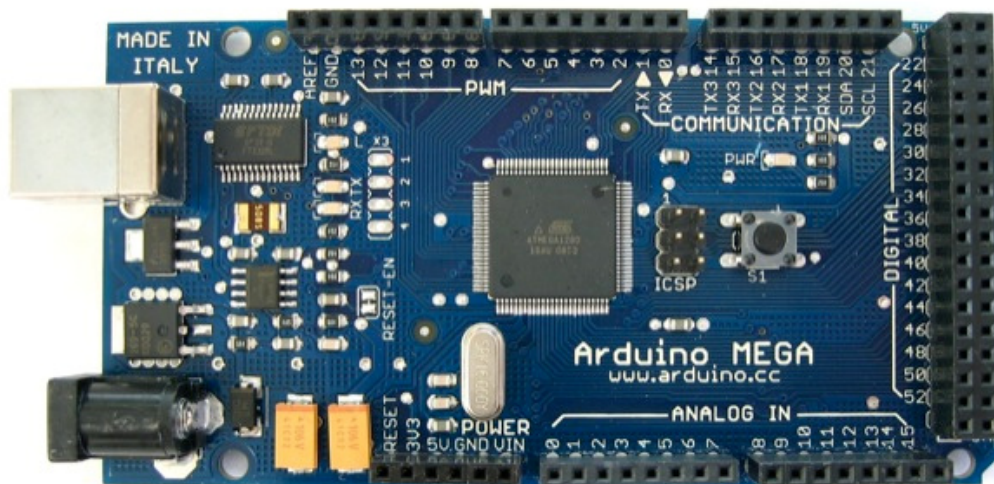

Figure 10. Picture of Used Arduino Mega Microcontroller

3. Three H-bridges, type L298.

4. Three stepper motors type PK246PB with step angle $\alpha_s = 1.8^o$

5. CNC kit, as shown in Figure 11.

6. Power supply AC220V, 50Hz, 4A, Adapter power 360W.

7. Cooling fan

8. Two leadscrews with pitch $p = 1mm/rev$.

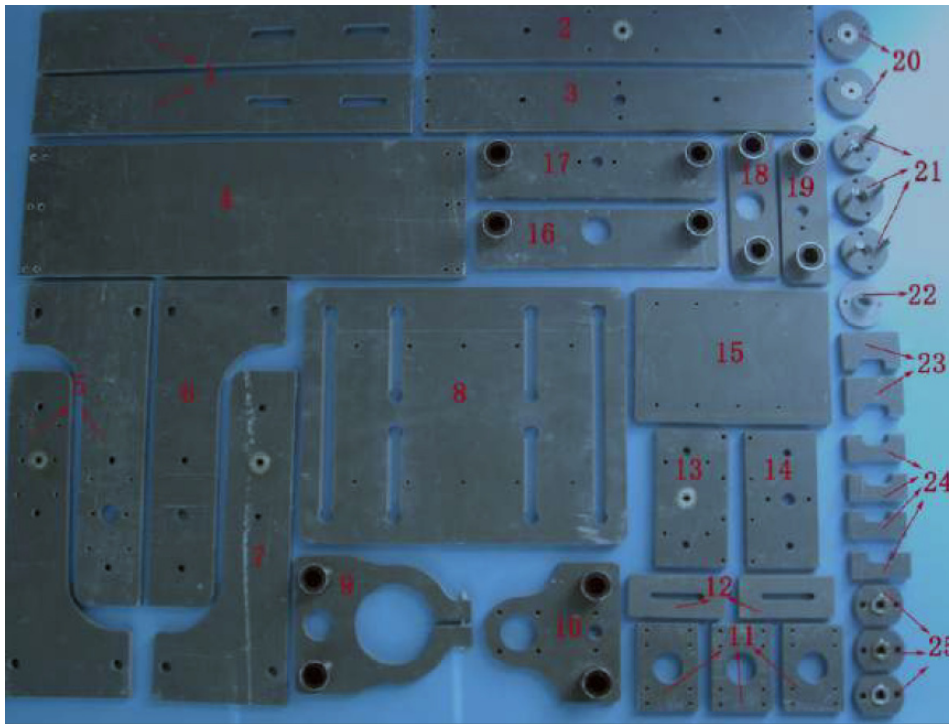9. Spindle motor, type domestic E240, $3000 - 12000 rpm$, $200W$

Figure 11. Picture of CNC Kit

Ignoring the standard deviation of the error distribution σ, the technical characteristics of the proposed CNC machine tool, based on open-loop positioning system using stepper motors, are given in Table 4.

Table 4. Technical Characteristics of Proposed CNC Machine Tool

Performance indicator	Calculation formula	Value
Control resolution of mechanical system CR_1	$CR_1 = \dfrac{p}{n_s}$	$0.005 mm$
Control resolution of MCU CR_2	$CR_2 = \dfrac{L}{2^b - 1}$, where L the range of axis and "b" is is the number of bits in the storage register for the axis.	$0.784 mm$
Control resolution of the machine tool CR	$CR = \max\{CR_1, CR_2\}$	$0.784 mm$
Accuracy of positioning system $Acc.$	$Acc. = \dfrac{CR}{2} + 3\sigma$	$0.392 mm$
Repeatability R	$R = \pm 3\sigma$	$Zero$

Two sets of experiments were conducted under certain machining conditions. The first set of experiments was performed under the following machining conditions: Rotational speed of stepper motor (leadscrew) =60 rpm, table velocity (feed rate) = 60mm/min, and pulse frequency = 470Hz. The results of the first set of experiments are shown in Table 5.

Table 5. Results of the First Set of Experiments

Task No.	Type of motion	Axis	Angular displacement, $A_m^{\ o}$	Direction	Programmed move, mm	Measured move, mm	Percentage error, ε
1	Straight line	X	10800	Forward	30	38	24%
2	Straight line	Y	10800	Forward	30	38	24%
3	Diagonal	X, Y	10080	Forward	28	32	14%
4	Straight line	X	3600	Backward	10	13	30%
5	Straight line	Y	7200	Backward	10	12	20%
6	Straight line	X	10800	Backward	30	36	16%
7	Straight line	Y	10800	Backward	30	35	16%

From the results of the first set of experiments it was found that the percentage error is relatively high and not suitable for positioning system. In addition, there is no mean to control the number of steps per revolution of stepper motors through Arduino software. Different attempts have been done to reduce the relative error of positioning system through Matlab software. The solution is to make a time delay when executing each pixel (equivalent to 1mm move) by 1.6 s for each pixel. This means that the machining time will be increased. The second set of experiments was conducted under the same conditions, as for the first set, but with time delay. The results are shown in Table 6.

Table 6. Results of the Second Set of Experiments

Task No.	Type of motion	Axis	Delay time, s	Programmed move, mm	Measured move, mm	Percentage error, ε
1	Straight line	X	10	6.25	6	4%
2	Straight line	X	30	18.75	18	4%
3	Diagonal	X, Y	10	14.14	13.5	4%
4	Diagonal	X, Y	30	26.52	25.46	4%
5	Straight line	Y	10	6.25	6	4%
6	Straight line	Y	30	18.75	18	4%

From the results in Table 6, it is clear that the percentage error has been significantly reduced.

4. Conclusion

In this paper, a novel technique based on the use of Matlab and Arduino software to control the CNC machine tool has been illustrated and experimentally verified. Experimental results show that the percentage error in positioning system can be reduced by using the Matlab software. The proposed technique can be used for machining and non-machining applications of NC.

References

Chang. I., Deng. J., Chan. S., (2000). A Next Generation Machining System Based on NC Feature Unit and Real-Time Tool Path Generation, *Int. J. Adv. Manuf. Technol.*, 16, 889-901

J. Balic (2006). Model of Automated Computer Aided NC Machine Tools Programming, *Journal of Achievements in Materials and Manufacturing Engineering*, 17, 309-312

Joze Balic (2004). Intelligent Computer Numerical Control Unit for Machine Tools, *Journal of Intelligent and Robotic System*, 40, 343-358

M. Tolouei-Rad, G. Payeganeh (2006). A Hybrid Approach to Automatic Generation of NC Programs, *Journal of Achievements in Materials and Manufacturing Engineering*, 14, 83-89

Pimoz Krzic et al. (2009). Step-NC: A New Programming Code for the CNC machines, *Journal of Mechanical Engineering*, 55, 406-417

S. Zietarski (2001). System Integrated Product Design, CNC Programming and Post Processing for Three-Axis Lathes, *Journal of Materials Processing Technology*, 109, 294-299

Yoham Kohen (1976). Interpolator for a Computer Numerical Control System, *IEEE Transactions on Computers*, 25, 32-37

The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage:
http://www.iiste.org

CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

Prospective authors of journals can find the submission instruction on the following page: http://www.iiste.org/journals/ All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: http://www.iiste.org/book/

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar