Autonomous Chess Playing Robot

Varun Gupta[1], Amit Kumar[2], Shilpi Agrawal[3], Saumya Jaiswal[4]
Robotics Club, Science and Technology Council
Indian Institute of Technology Kanpur
Kanpur, India

***Abstract*— Autonomous Chess Playing Robot is a robot that can challenge a human's chess playing ability in a tangible environment with its varying difficulty level. As any other autonomous robot, ACPR is based upon three pillars, viz. electronics, programming, and mechanical design. The robot consists of a customized chess board below which LDR sensors have been fabricated on a PCB (Printed Circuit Board). Multiplexers also being laid on PCB provide a way for one-way serial communication between sensors and microcontroller. Linear sliders facilitate horizontal and vertical movement of the gripper to move a piece from one position to other. The linear sliders use rack and pinion mechanism while the gripper uses the concept of four bar linkage mechanism. The crucial element of the robot i.e. the intelligence has been incorporated with the help of MinMax algorithm supported by alpha-beta pruning. The microcontroller is connected to a computer terminal by a USB and the chess engine code runs on the computer terminal itself.**

Keywords*— Robot, Chess, Artificial Intelligence, Rack and Pinion, Microcontroller, Arduino, Multiplexer, Motor Driver, LDR sensor, Linear Slider, Gripper, Four bar linkage mechanism, MinMax algorithm, alpha-beta pruning, etc.

I. INTRODUCTION

The aim of project ACPR is to make a robot that can play the game of chess against human on standard size chess board and pieces. The game of chess is a two player strategy board game played on a chessboard, a checkered game board with 64 squares arranged in an eight-by-eight grid. It is one of the most popular games; played by millions of people worldwide. Normally the game of chess is played between two human players involving an intensive exercise of mind on both sides. Our project tries to replace one of the human players to shape in a robot vs. human challenge.

The robot requires a lot of perfection in terms of precision, identification and intelligence. Even an error of 1mm in the movement of the mechanical model successively may give rise to an error as big as one-eighth the size of the board. The accurate detection of color, type and position of the chess piece is a necessary condition for smooth functioning of the complete game of chess lasting for 45 minutes on an average. At the same time, the robot should be made enough intelligent to challenge a human.

The task of one move by robot can be classified into three parts viz. identifying the current status on the chess board, checking for any incorrect move for human, calculating the best possible move for the robot and triggering the mechanical structure to implement this move. The electronic circuitry comprising of LDR, multiplexers and microcontroller identifies the current status of the chess board. The decision for the best possible move is taken based upon artificial intelligence implemented by the min-max algorithm supported by alpha-beta pruning.

This paper is organized as follows:

- Section 2: Mechanical design
- Section 3: Electronics
- Section 4: Programming
- Section 5: Conclusion
- Section 6: Acknowledgment
- Section 7: Future Scope
- Section 8: References

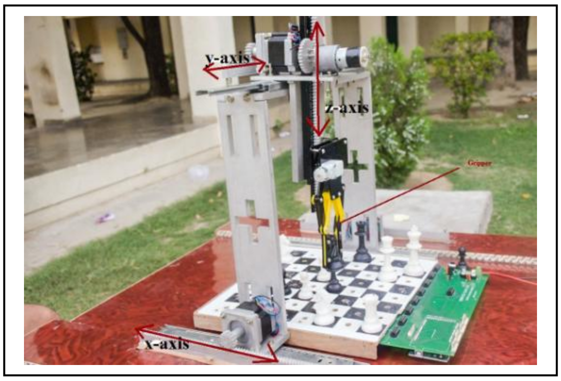

Fig 1: Photograph of the robot

II. MECHANICAL DESIGN

The mechanical design is a substitute of the physical aspect of the second human. Our design is a simplified version of a human hand equipped with four degrees of freedom. Similar to the human hand the mechanical model should perform the task of moving a piece from one position to the other. This part is the crucial aspect of the project and no significant error is desirable. The mechanical design can be classified into two parts:

- Linear x-,y-,z-axis movement
- Mechanical Gripper

The size of the chess board is 12’12’’ and each square block has 1.5’1.5’’ dimension. During making of a turn by the mechanical gripper, the movement can be expressed as a linear combination of four independent variables. The variables are x, y, z and gripping extent.

A. *Linear three axis movement*

We implemented the three-axis movement with the help of linear sliders which are based on rack and pinion mechanism. The rack being fixed with the base is stationary.

The 'pinion' a wheel gear is connected with a motor. The pinion meshes with the toothed rack and thus rotary motion is converted into reciprocating motion. There are two stepper motors being employed for x-axis movement and one stepper motor for y-axis movement. The stepper motors have a precision of 1.8 degrees. For z-axis movement, we have used a high torque DC motor with encoder which operates on a rack fixed vertically to the top bridge structure. We have used high torque DC motor so as to counter balance high torque by a chess piece weight and friction. The stepper motors and DC motor movement is operated by a microcontroller. The vertical supports have been mechanically fastened by screws strong enough to circumvent any possible toppling due to the torque of hanging structure of gripper and rack-pinion structure.

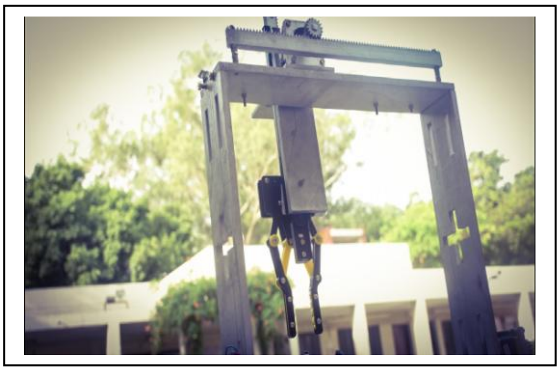

Fig 2: Mechanical Model of the Robot

B. *Mechanical Grippers*

The task of the gripper is to grab the chess piece for moving it from one position to the other. The gripper starts functioning as soon as it starts to hover over the chess piece desired to be moved. It consists of a frame which is based on four bar linkage mechanism. The stepper motor is driven according to the instructions it receives from the Arduino.

III. ELECTRONICS

Electronic parts of this robot are microcontroller, motor driver, stepper motors, DC motor, LDR, multiplexer ICs. Microcontroller is intellectual center for the robot. We have used "Arduino Mega" which is a microcontroller board based on the ATmega1280. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports). Microcontroller detects the status of the chess board using LDR sensors installed on PCB (Printed Circuit Board) fixed beneath the chess board and sends the information to a computer terminal. The terminal sends instructions to the microcontroller for the next move and in return the microcontroller commands the motor driver. The electronic part of the robot can be classed into three broad divisions:

- Chess piece position mapping
- Serial Communication between microcontroller and computer terminal
- Motor movement control

A. *Chess Piece Position Mapping*

The logic for detection of chess piece position is: we can deduce the position of each piece after the nth move by simply comparing the position of pieces after the (n-1)th move and the position of undetermined objects after the nth move (hereafter, an object will designate a chess piece whose type hasn't been recognized yet, whereas a piece refers to a chess piece whose type is known).

We have employed LDR (light dependent resistors) sensors for detection of any chess pieces' position on the chess board. LDR is a variable resistor which changes its value based upon the availability of light. In normal room light conditions, the resistance is typically 8k ohms and in dark condition the resistance amounts to more than 100k ohms. We have used the LDR in a potential divider circuit with a resistor of 40k ohms connected in series and a power source of 5V. When ambient light falls on LDR, voltage drop across it is 1.5 to 2V and in dark condition, the potential drop is close to 5V.

The designed chess board consists of sixty-four holes below which LDRs are soldered on PCB. Whenever there is a chess piece on any square block, the LDR beneath it faces completely dark condition and in the absence of chess piece it receives light. The voltage across the LDR is sent to the microcontroller in digital format through double level of multiplexing. The chess board can be split into eight rows and for each row we have used 8:1 mux. The outputs of eight mux (for eight rows) are connected to the ninth 8:1 mux. The output of the second level multiplexer (the ninth mux) is connected to the input of the microcontroller. By varying the data selector pin values of nine multiplexers, we access the status of each square block serially and this information is subsequently sent to the computer terminal.

The HIGH input to the microcontroller implies the presence of chess piece on the square block and vice versa.

We have used multiplexer IC SN74LS151N to curtail the number of pins required for taking the inputs. Data from sixty-four square blocks is sent serially to the microcontroller by controlling the data selector pins of the multiplexers.

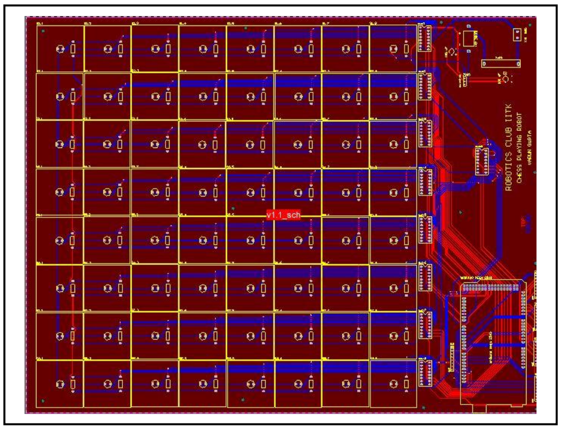

Fig 3: Printed Circuit Board Layout

B. *Serial Communication*

The microcontroller is connected to the computer terminal by a USB [Universal Serial Bus]. The microcontroller sends information to the terminal in bit by bit manner which is essentially a serial communication. It involves the use of Rx and Tx pins of the microcontroller. Sixty-four bits in the form of 0 and 1 is sent to terminal and terminal uses this information to decide the next move for the robot based on a C program. The next move information is serially communicated to the microcontroller.

C. Motor Movement Control

The microcontroller is connected to motor drivers which act as control system for the motors. The microcontroller provides a supply of 5V which is not sufficient to drive motors. An external supply of 12V is provided to motor drivers to run the motors. Motor movement involves clockwise and anticlockwise movement of the stepper motors and dc motor.

IV. PROGRAMMING

Programming is one of the most important part of the project as this is the way of making robot think and exhibit intelligence. Programming involves microcontroller coding, programming implementation of serial communication and developing chess engine.

A. Chess Engine

Through chess engine code, the robot is able to think the next best possible move given the human completed his/her turn and the status of the chess board. We developed the chess engine in C++ and its level of difficulty could be varied as desired.

Our chess engine is primarily based on MinMax algorithm. MinMax algorithm is used in two player games which are zero sum games. Zero sum game means the sum of score of two opponents is zero. One player is assigned positive value and other player a negative value which means a player's loss implying other player's gain. The task of each player is to maximize its absolute score.

To evaluate the best possible move for the robot, we take all the best possible moves by robot and then all possible moves by human and proceed in a similar fashion to a certain depth level and thus constructing a min-max tree. Here is an account of the complexity of the chess engine code.

On an average, there are twenty possible start moves and twenty possible replies. In this way, there become four hundred possible positions after two ply [half move]. In this way there is an exponential explosion and the number of possible combinations after ten plies [5 robot and 5 human moves] becomes more than a billion.

To calculate the best possible move, we check for any checkmate and if not we evaluate the leaf node score. The score is calculated based upon the numerical weight of each kind of chess piece. The robot's pieces are assigned positive value and the human's pieces are assigned negative values.

- Pawn: 10
- Knight: 31
- Bishop: 33
- Rook: 50
- Queen: 79
- King: 10000

Score at every leaf node is evaluated as:

Score = (sum of all white piece value present on chess board [a positive number]) + (sum of all human piece value present on chess board [a negative number])

The complexity of MinMax algorithm is O(b^d) where b is branching factor (~ 40) and d is the number of plies ahead that we want to search. Seeing the tree size it is very crucial to use some kind of pruning. We have used alpha-beta pruning which significantly reduces the time required to calculate the best possible move.

The basic logic of alpha-beta pruning is that if we know for sure the value of our result in any search is more than x and the current search result for this branch so far can give no more than x, then there is no need to search this branch any further.

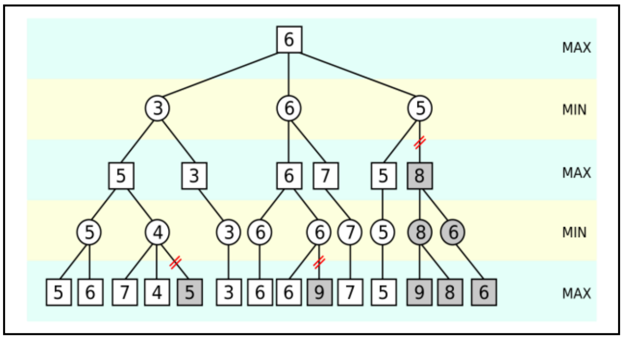

Figure 4: An illustration of min-max search tree and alpha–beta pruning. The grayed-out sub-trees need not be explored (when moves are evaluated from left to right), since we know the group of sub-trees as a whole yields the value of an equivalent sub-tree or worse, and as such cannot influence the final result. The max and min levels represent the turn of the player and the adversary, respectively.

B. Microcontroller Coding

The basic purpose of microcontroller coding is to receive inputs from LDR sensors on the chess board, send and receive information to/from the computer terminal and give instructions to motor driver for the motor movement control.

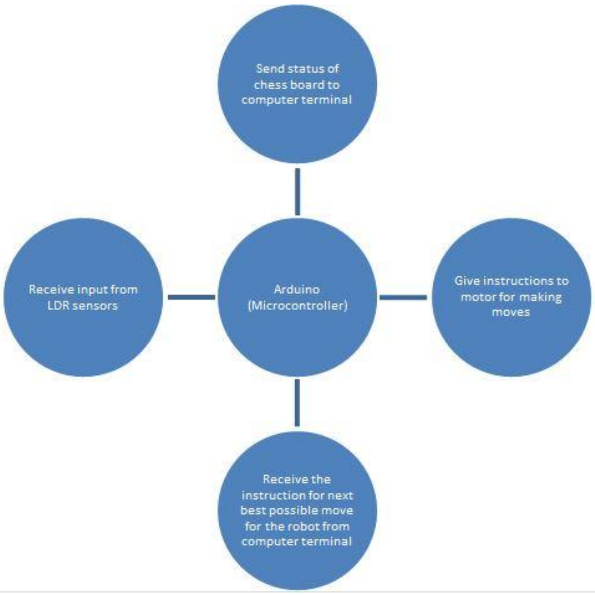
Send status of chess board to computer terminal
Receive input from LDR sensors
Arduino (Microcontroller)
Give instructions to motor for making moves
Receive the instruction for next best possible move for the robot from computer terminal

Figure 5: Working of Microcontroller

V. CONCLUSION

It is not wrong to say that the future is the era of robot civilization where human and robots will interact more intimately and frequently. We are hopeful that artificially intelligent robots will find a prominent place in military, government, corporate world, education centers, entertainment industry, etc. The autonomous chess playing robot with primary level of artificial intelligence is one step in this direction.

In this paper, we have discussed one method of making an autonomous chess playing robot. We are successful in playing games with this robot. This robot is kept in the Student Activity Center, IIT Kanpur for public showcase.

VI. FUTURE SCOPE

Robotic revolution is going to come in the future. And the autonomous chess playing robot certainly has a good future in the entertainment sector.

One can use this robot to hone his/her chess playing skills. The main advantage is the provision for adjusting the level of difficulty. The robot provides an excellent physical environment for smooth conduction of the game of chess. This robot immensely contributes to the robotic entertainment industry.

VII. ACKNOWLEDGMENT

The satisfaction that accompanies the successful completion of this project would be put incomplete without the mention of the Robotics Club and people who made it possible, whose constant guide and encouragement crown all the efforts with success.

We would like to thank our parents - ‘the first and most important teachers’ - who motivated, guided and supported in every possible way.

We would like to express our deep sense of gratitude to the Robotics Club, Science and Technology Council, IIT Kanpur for providing the atmosphere to learn and innovate and for financial support.

We would also thank the Club coordinators Akshay Massare, Deep Goyal, Rishi Gupta and Shubham Patel and also the mentors Hardik, Mukund and Zaid who made valuable suggestions throughout the project.

We would also like to thank our friends who helped in little-little ways directly or indirectly. We also thank the staff of 4i laboratory and Tinkering laboratory of IIT Kanpur for providing us the facility to develop mechanical model.

VIII. REFERENCES

[1] Abhishek Attal, Diptiman Purbey, Aman Khatri, Kushang, Abhishek Kumar “Piano Playing Robot,” International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181. Vol. 3 Issue 5, May - 2014

[2] Wikipedia

[3] Tutorials of Robotics Club IIT Kanpur

[4] Arduino

[5] COUR, Timothée, Rémy LAURANSON, and Matthieu VACHETTE. "Autonomous Chess-playing Robot." Ecole Polytechnique, July (2002).

[6] http://www.iis.sinica.edu.tw/~tshsu.