AI 연구 플랫폼을 위한 저비용 체스 로봇 설계

Sehar Shahazad Faraooq ᴼ Hafiz M W Khalil* Adeel Arif**

세종대학교 컴퓨터공학과/세종대학교 물리학과*/Bermen 응용과학대학 전기전자공학과**

sehar146@gmail.com, wasi2k4@gmail.com, adeel.hammad@gmail.com

Cost-Effective Design of Autonomous Chess Playing Robot for AI Research Platform

Sehar Shahazad Faraooqᴼ Hafiz M W Khalil* Adeel Arif**

Dept. of Computer Engineering, Sejong University/Dept. of Physics, Sejong University*/Dept. of Electrical and Electronics Engineering, University of Applied Science, Bermen, Germany**

Abstract

This paper present an intelligent microcontroller based chess playing robot which can play a board game against opponent and calculate its moves in a non-idealized environment. In this work, for the sake of simplicity task is accomplished by using Cartesian coordinate system. Chess playing system has been designed in such a way that it provides an interface between user and robot to control chess movements using RS232. Various algorithms are implemented for interfacing hardware in C++ language. Our main goal is to design a cost effective and highly accurate robot system that consumes minimal power to complete its task.

1. Introduction

The main concern in robotic chess playing system is a physical board game due to their adjustable degree of structure because of involving board, pieces, state and their manipulation. Many schemes have been proposed to achieve accurate tracking control for non-linear systems. The problem is to make an exact robotic model with perfect measurements. Although many control schemes have been proposed to track the robot but most of them lacks in determining its precise location [1]. Recent, experiments suggest that the limitation of locating a robot more precisely can be resolved by using digital observers. Hence, digital observer is a good solution to obtain collective response about the given robotic model. Furthermore, most of the existing work employed coordinate system with servo motors to design the robotic system. However, the accuracy achieved using joint coordinate system and a servo motor is not satisfactory. Therefore, in this paper, we designed an Autonomous Chess Playing Robot (ACPR), which gives more exact and precise measurement using Cartesian coordinate system. Moreover, our developed system is made up of stepper motors rather than servo motors. The results of our developed system manifest its effectiveness to be used as a robotic control mechanism.

The remaining paper is organized as follows. In Sec. II, the mechanical design of ACPR is discussed, Sec. III gives the detail of the electrical design of ACPR, Sec. IV explains the driving software for the ACPR and Sec. V includes the conclusions and the future work.

2. Manipulator and Sensing Suite Design

Usually physical board game requires a number of instruments and pieces to make the perception and manipulation simpler. However, the handlings of complexities related to these arbitrary pieces are quite challenging and includes lot of concerns [2]. Our Robotic system includes a completely new design to handle such complexities. Several factors were considered to make, a low cost, power efficient, fast, accurate and user friendly system for both the hardware manipulation as well as measurement of the motions.

2.1 Mechanical Design

Figure 1. ACPR Mechanical Model

To achieve accuracy and efficiency in performance, the idea of using Cartesian coordinates system for the control system is quite well known. Therefore, we opted it to use for our developed robotic system. Figure 1 demonstrates the environment i.e. is a chess game to evaluate our developed system. The environment, in which our developed robotic system is deployed, consists of a space that can be represented using Cartesian coordinate system consisting of three axes X, Y and Z.

A panel is mounted on each axis, denoted by X, Y and Z panels respectively. The movement of each axis is controlled by a stepper motor. The Y and X panel can move to and fro whereas Z panel can move up and down. A Gripper is attached to the bottom of the Z panel.

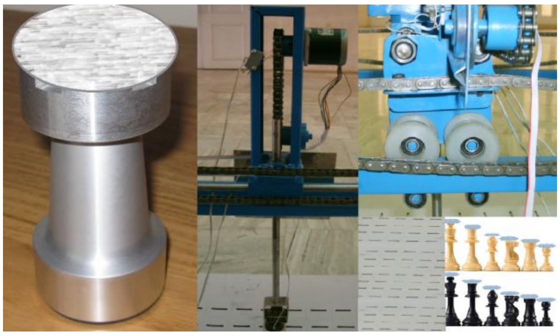

Figure 2. ACPR Chess piece design, the gripper, plastic wheel and bearings geometry, Sensor placement in chess board, chess pieces with different size and equal weight.

To calibrate the Z-panel steel rods are fixed on the Y panel. Plastic wheels with bearings are used in order to allow continuous motion of the panels. The whole systems including the chessboard as well as panels are enclosed in a metallic frame. To avoid collision of panels during their movements, read switches are located around the frame for safety. The length of a single square box of the chess board is 5cm so the total movement area for X and Y panel is restricted to 50 cm. The chessboard playing area consists of eight rows and eight columns. But a spare space is also accessed by the moving panels to throw the capturing piece outside the chessboard. While the Z panel movement in down direction is controlled on the basis of the size of the chess pieces and its initial position is near the read switch at the top of the z-axis. Chess board is made in such a way that a weight switch is placed at the bottom of each square of chess to get information of which square box is empty or occupied. An empty box has zero weight whereas occupied with 170 g. The identification of black and white boxes in chess game is memorized in the algorithm just to avoid any visionary implementations which require more power consumption.

2.2 Chess Pieces and Magnetizer

The size of chess square is increased to hybridize weight sensors within the boxes. It will also help to make considerable calculations. The chess pieces are designed in such a way that a round metal piece is fixed at the top of the Chess piece and a magnet is placed at the bottom of the piece as shown in figure 2. The diameter of metal plate and magnet is 2.5cm and weight of each piece is fixed to 170g to avoid the use of multiple grippers with different magnetizing power. In order to pick the chess pieces by electromagnet, there is a circular coil at the bottom of the gripper. An electromagnetic gripper is used to pick and drop the chess pieces. At the centre of the electromagnet, a very tiny push switch is used to stop the movement of the Z panel towards the top of the chess piece. This switch provides information of either the chess pieces is picked or missed. If the chess piece is missed by any reason, the gripper moves down again to pick the piece. The robot terminates the game if the chess piece is not picked in three trials. This problem can arise due to number of reasons such as the misplacement of the chess piece, an opponent making illegal move physically or by occluding the pieces.

2.3 Sensing

To make the board sensible, the weight switches are placed at the bottom of each box. These switches provide information of box state (empty or occupied). This information is used through the rest of the game.

3. Electronic Design

Table 1. System Specifications

Motor Type	DC Stepper motor(86BYG)
Step Size	1.8 degree (200 step per rev)
Motor Driver IC	L293D [3]
Power supply	24V DC, 12V DC, 5V DC
Max. Current	1A
Micro controller	ATMEL's AT89C51 [4]
Interfacing port	RS232 Serial port
Interfacing IC	MAX232 [5]
Read switches	MDS003C
Weight switches	MDSM-4R 12-18
Magnetizer limit switch	Miniature pushbutton

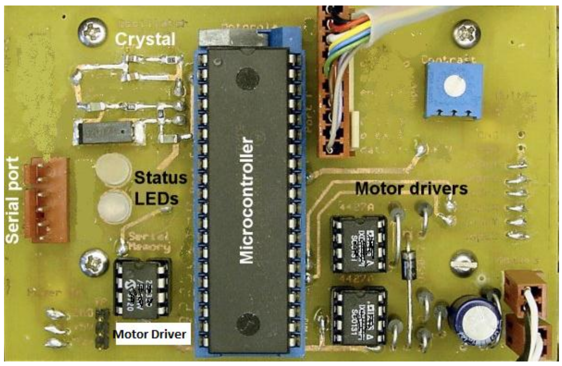

Figure 3. Electronic Circuit

Any turn taken by the user from computer need implementation. This is done by communicating serially to the robot using controller circuit. The controller circuit consists of microcontroller, motor driving IC and sensing network as shown in figure 3. The specifications for the electronic circuit are given in the table 1.

4. Driver Software

The driver software for ACPR is made in C++ programming language and the communication between the computer and the ACPR is done by serial port. The display of the whole chess game is presented in computer and mouse is used to play chess in the software. The basic initial stage of the chess is predefined in the program file named **Chess library** and each turn of the user as well as the ACPR is saved in the **Chess polo** file. This storage pattern reflects no errors in ACPR. Any wrong turn is indicated by the program and is not allowed to process. The move taken by the user is first implemented by the ACPR using simple displacement formula by determining the coordinates of the destination to source chess boxes from **find me here** file and then new value board array of destination box are stored in the **Chess polo** file to remember the move of the user. After the user's move is made, the **find me here** file is also updated with new coordinates of the latest chess square. Finishing user turn, ACPR reacts affectively to the status of the game and the action of the opponent. It thinks the best possible move and proceeds to new directions by re-calculating the value board array values. For the ACPR to complete its move, user has to wait for its turn during which the user can listen a music file as hold.

4.1 Playing Chess

ACPR plays a game against the opponent with fully predefined rules. It doesn't make any illegal move neither allow the opponent to go for it. The initial configuration of the chess should be the same as defined by the algorithm to identify the chess pieces. The black queen and king should be placed at d8 and e8 while rooks, knights, and bishops at a8,h8, b8,g8 and c8,f8 respectively. The eight pawns are placed next to the base row from a7 to h7. Similarly white chess pieces specified as robot assets are placed at the opposite side. The queen is placed at e1, king at d1, bishops at c1,f1, knights at b1,g1, Rocks at a1,h1, and the eight white pawns from a2 to h2.

The movements of all the chess pieces are defined in the algorithm. Priorities are given to the chess pieces depending upon their power and moves such that at any situation it is tried to avoid castling. Both the defending and the attacking situations are specified in the software. The second highest priority after the king is given to queen due to its highest power in chess. It can move anywhere in any direction and without the restriction of no. of boxes for its turn, so to attack from the queen or to defend the queen is better than to use a pawn or knight. The opponents highest power is measured and tried to capture the most powerful chess piece by making a situation to force the user to go for lose move.

For any move by an opponent, ACPR first check either it's legal or not. If legal then the coordinates of the move are calculated. The control signal and directions to motors are provided to reach the box. The X and Y panel move together to save time. The gripper picks and drops the chess piece from source to destination box. When the user's move is finished, ACPR thinks a move against the opponent. The same procedure is repeated for its own turn. Instead of defining an initial state of motors and restricting the motors to reach to their initial states after each turn the latest position state of motors are updated to the software buffers to save time. To capture the opponent's piece either by the user or by ACPR, first captured piece is placed outside the chess then capturing piece is moved to desired location. The completion of the move is identified by the weight cells at the bottom of chess boxes provide information of the source box is empty and destination box is occupied. ACPR only remembers the initial state of chess and each move taken by any chess piece instead of identification of chess pieces. The problem may occur when the chess pieces are dislocated or placed manually. In that case the software couldn't find the problem and game is terminated. The average time taken by the ACPR to move the user's turned and then by its own turn is measured as 35 seconds and the average time when a capturing move is carried out is 47 seconds. This time is measured again and again and the reason of consuming more time in capturing move is because the robot has to perform three tasks, first to clear the attacked box, second to put the attacking piece to that box and thirdly performing its own turn.

5. Conclusions and Future work

This paper described the implementation of playful application to explore the human-robot interaction. Our main goal is to build a low cost and high level design of intelligent robot to play chess game against user in real time scenario. This robot is consisting of three main functionalities: recognizing chess players, computing new moves and executing these moves by means of Z panel which has electromagnet to pick the pieces. To improve the accuracy of the developed robotic system, a concave bowl shaped plastic cover is used around the electromagnetic gripper to reach exactly to the top of chess piece. *The total cost of the ACPR is $ 800.* In future the wired data transmission can be replaced with wireless protocol such as Zigbee or WiFi and electromagnetic gripper with jaw.

6. Acknowledgement

This research was supported by Basic Science Research Program and the Original Technology Research Program for Brain Science through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0001749) (2012-0005799). Special thanks to Professor KyungJoong Kim for his support and motivation to complete this work.

7. References

[1] C. Breazeal, A. Brooks, D. Chilongo, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, and A. Lockerd, "Working Collaboratively with Humanoid Robots," in Humanoids, 2004.

[2] J. T. Feddema and O. Mitchell, Vision-guided Serving with Feature based Trajectory Generation for Robots, IEEE Trans. on Robotics and Automation, vol. 5, no.5, pp. 691–700, 1989.

[3] IC-L293D , http://www.datasheetcatalog.com/datasheets_pdf/L/2/9/3/L293D.shtml

[4] Microcontroller ATMEL AT89C51, www.atmel.com/atmel

[5] IC-MAX232 http://www.datasheetcatalog.org/ datasheet /texasinstruments/max232.pdf