

# Polar and Polygon Path Traversal of a Ball and Plate System

Aneeq Zia

*Electrical Engineering Department, LUMS School of Science and Engineering*

*D.H.A, Lahore Cantt, 54792, Pakistan*

aneeq91@hotmail.com

**Abstract**— This paper builds on the traditional ball and plate balancing system; a novel method for making the ball follow the path of any polar graph, with the equilibrium position of the plate as the origin, is discussed and extended to make the ball follow any ‘n’ sided polygon. A 5-wire Higgstec touch panel coupled with Arduino Duemilanove with Atmega 328 was used. A description of the hardware and software used and the algorithm followed is also presented.

**Keywords**— ball and plate, control, polar, polygon, touch screen, Arduino

## I. INTRODUCTION

Ball and plate system is a nonlinear multi-variable system. It's a two-dimensional electromechanical device as an extension of the ball and beam system [1].

A steel ball on the hard plate is an unconstrained object which is able to move freely and has no ability to recognize the environment. So the unconstrained object cannot control its behaviour by itself. This makes motion control of ball and plate system difficult. Since the system is inherent nonlinear, the mathematical model is difficult to be derived [2].

Various control schemes for the balancing of the ball have been devised. Out of these, one of the most effective schemes is the double feedback loop structure [1]. The inner loop serves as a dc motor servo position controller, and the outer loop controls the ball position.

This paper describes the control required for balancing and making the ball follow any random polar path or an ‘n’ sided polygon using Arduino Duemilanove with Atmega 328. It extends the conventional ball and plate balancing systems to more complex problems of path traversal by the ball. A detailed description of the system and the associated control used to make the ball follow a generalized polar path follows. The results were then used to make the ball follow any ‘n’ sided regular or irregular polygon. The rest of the paper is organized as follows. Section 2 gives the details of the controller board and the software used. The mechanical description of the ball and plate setup used is described in Section 3. Section 4 deals with the details of the touch panel used. The explanation of the traversal of the ball on a general polar path is discussed in Section 5. Section 6 extends the discussion of Section 5 and explains the mathematics involved and the implementation of a method for traversing a regular

polygon by the ball. Path traversal of any irregular polygon is dealt in Section 7. Conclusion is made in Section 8.

## II. THE BOARD AND SOFTWARE

The Arduino Duemilanove is a microcontroller board with either ATmega168 or ATmega328. The latter one was used for the development of the control for the system. It has a 16MHz crystal oscillator, 6 analog and 14 digital input output pins. A universal serial bus connection was also available to burn the code onto the controller and for serial communication. Arduino IDE was used for compiling the C/C++ code and burning it onto the controller. Proteus VSM, Virtual Breadboard and MATLAB were used for preliminary simulations.

## III. MECHANICAL DESCRIPTION

Two hobby servo motors were used for the 2-axis motion of the touch screen. Each of the servo motors drives one axes of the plate-rotation angle. A spatial linkage mechanism was used to connect the plate with the motors. A ball and socket joint was used for pivoting the plate and touch screen from the equilibrium point. As the touch screen was rigid, the plate on which the touch screen was mounted was made as light as possible in order to have a better response time. The linkage of motor with the screen ensured that the motor angle and plate angle can be safely assumed to be equal.

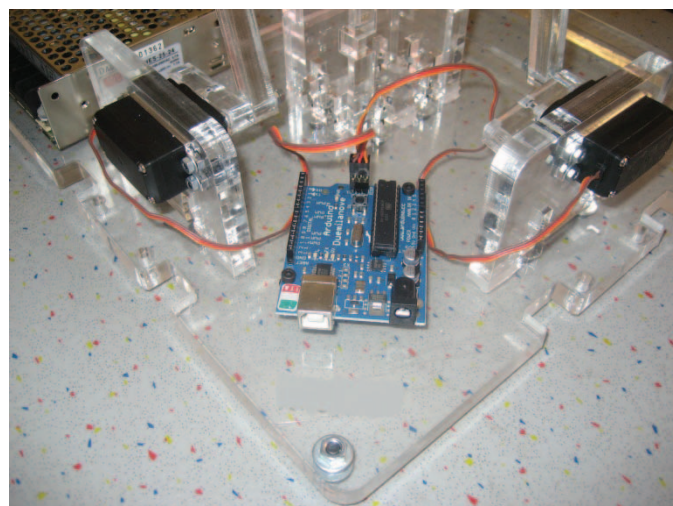


Fig. 1 The Arduino Duemilanove together with its connections with the servo motors

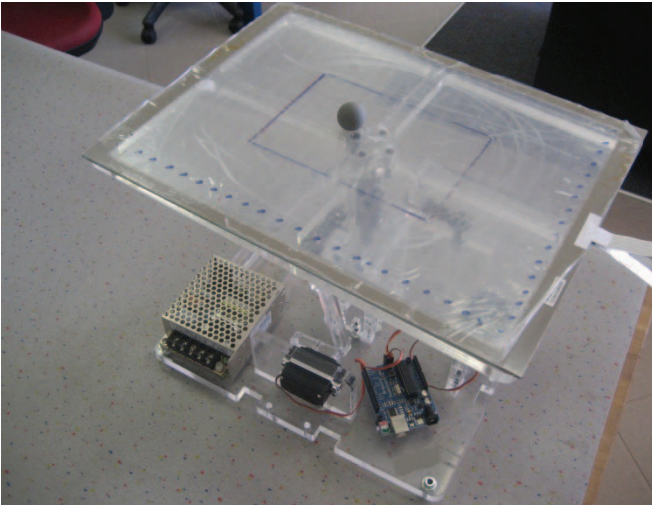


Fig. 2 Ball on plate physical setup

#### IV. TOUCH SCREEN

HIGGSTECH 15.4" five wire resistive touch screen was used to measure the position of the ball. The response time of the touch panel, according to the manufacturer, is 20msec which was acceptable for this experiment. Four of the wires of the touch panel were connected with the four corners whereas the fifth one was used to obtain the analog signal indicating the position of the ball. The contact point is measured using the homogenous resistive surface. An even voltage gradient appears across the resistive surface when a certain potential difference is applied to the electrode pair. Another conductive layer is required to carry out a high-resistance voltage measurement. Hence, the touch screen can be considered as a pressure controlled electric switch.

The analog signal from the touch screen ranged from 0V to 5V when a potential difference of 5V is applied in one direction. As the touch screen had only one output pin or the sense pin, both x and y coordinates could not be obtained simultaneously. The configuration of the touch screen used is shown in Table 1.

TABLE I  
PIN CONFIGURATION OF THE TOUCH PANEL

Mode	LT	RT	LL	RL	Sense
X-coordinate	Ground	Vcc	Ground	Vcc	Hi-Z/ADC
Y-coordinate	Ground	Ground	Vcc	Vcc	Hi-Z/ADC

LT, RT, LL and RL in Table 1 represent the wires connected to the left top, right top, left lower and right lower of the touch screen respectively. As can be seen in Table 1, the logic signal going to RT and LL had to toggle to measure both coordinates. This was taken care of in the algorithm.

The analog signal from the touch panel was converted to digital using the 10-bit analog to digital converter in the Arduino board. This means that the analog values read, which ranged from 0 to 5V, were converted to digital values between 0 and 1023. This gave the resolution of 5V/1024 units or 4.9mV per unit which was satisfactory for this purpose. By taking different measurements of the coordinates of points on the touch screen and the desired servo motor angles (-90 to +90) in order to balance the ball on the touch panel, a function for the relationship between the x/y coordinates and the corresponding motor angle was found using the curve fitting tool of MATLAB. These relationships are given in (1) and (2).

$$\theta(x) = 0.2706x - 48.188 \quad (1)$$

$$\theta(y) = 0.2609y - 39.556 \quad (2)$$

Fig. 3 and Fig. 4 show the curves obtained from the curve fitting tool of MATLAB. Using (1) and (2), the required angles for the servo motors were found according to the place of the ball on the touch panel. This strategy was used to balance the ball on the equilibrium point. Moreover, the ratios  $\frac{\theta_x}{x}$  and  $\frac{\theta_y}{y}$  were found by dividing the range of motor angles by the range of coordinates and are given in (3) and (4).

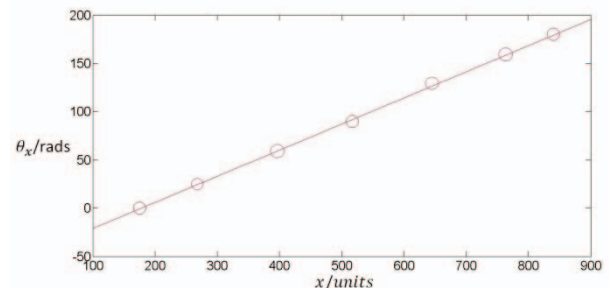


Fig. 3 Relationship between  $\theta(x)$  and x

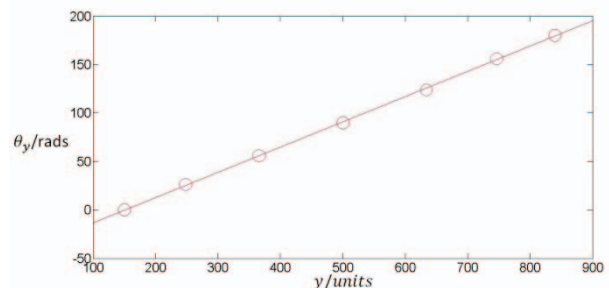


Fig. 4 Relationship between  $\theta(y)$  and y

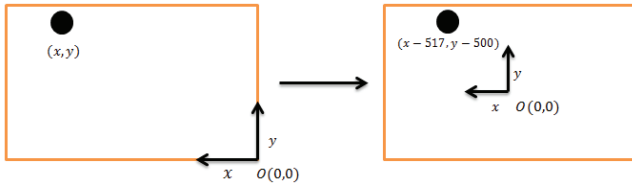


Fig. 5 Shift of coordinate system

$$\frac{\theta_x}{x} = 0.2707 \quad (3)$$

$$\frac{\theta_y}{y} = 0.2609 \quad (4)$$

Using (3) and (4), the appropriate motor angle for specific  $x_{difference}$  or  $y_{difference}$  was found. With these basics established, the rest of the discussion is focused on moving the ball on any polar graph or on the sides of any 'n' sided polygon.

## V. MOVEMENT ON POLAR PATH

The coordinate system was shifted so that the origin lied at the center of the touch panel. This is shown in Fig. 5. As the ball could never be precisely placed on the path, the ball had to be moved to the path at the start. Let's say that the polar graph to be followed is given by (5).

$$r = f(\theta) \quad (5)$$

The ball's present  $(x, y)$  coordinates obtained from the touch screen were converted to polar coordinates  $(r, \theta)$ . After doing this, polar coordinates of the point to be reached by the ball i.e. point on the polar graph having the same  $\theta$  as the ball's present position, were found. Value of  $r$  for the new coordinates was calculated using (6).

$$r = f\left(\tan^{-1}\left(\frac{y}{x}\right)\right) \quad (6)$$

The polar coordinates for the final point were then converted back to Cartesian coordinates. After having the final and initial points, the difference between the coordinates was calculated using (7) and (8).

$$x_{difference} = x_{old} - x_{new} \quad (7)$$

$$y_{difference} = y_{new} - y_{old} \quad (8)$$

Now that we have the difference in the two coordinates, the angles that motors need to rotate can be easily found using (9) and (10).

$$\theta_{\Delta x} = x_{difference} * \frac{\theta_x}{x} \quad (9)$$

$$\theta_{\Delta y} = y_{difference} * \frac{\theta_y}{y} \quad (10)$$

The loop for moving the ball to the desired path was continued until the ball was on or very near the desired path. This is shown in Fig. 6. A direction integer was used to manoeuvre the clockwise or anticlockwise movement of the ball along a path.

In order to move the ball on the path, similar logic to that of moving the ball to the path was used. The only difference was that the final coordinates were now given by (11) and (12), where  $d\theta$  represents the incremental angle used to move the ball on the polar path. This is shown in Fig. 7.

$$x_{new} = r(\theta + d\theta) \cos(\theta + d\theta) \quad (11)$$

$$y_{new} = r(\theta + d\theta) \sin(\theta + d\theta) \quad (12)$$

Once the ball had covered  $d\theta$  on the path, its position was checked again to ensure it had not "de-tracked". In case it had, appropriate measures were taken to bring the ball back on the path. See Fig. 8 for a detailed flowchart of the algorithm used for the traversal on a generalised polar path.

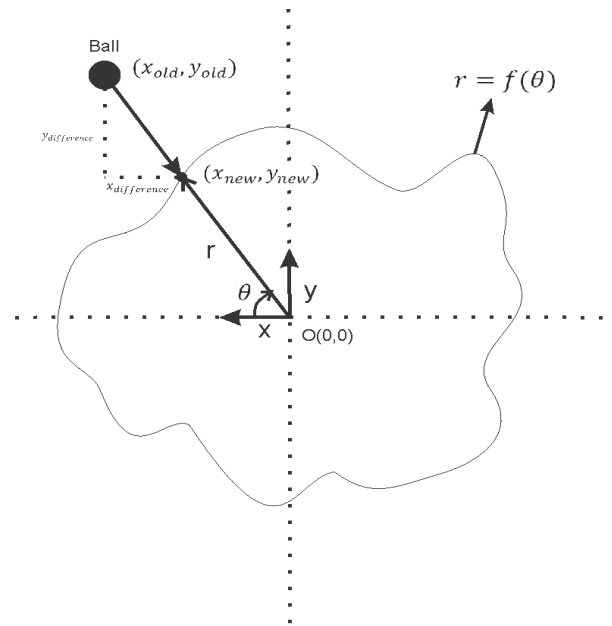


Fig. 6 Movement of the ball towards the path

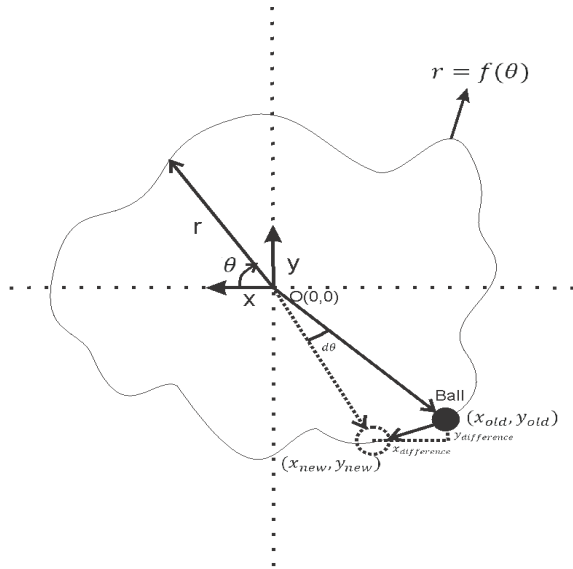


Fig. 7 Movement of the ball on the path

Now that the ball can move on any polar graph, the discussion is extended to make the ball move on the sides of any ‘n’ sided regular polygon using the same algorithm.

## VI. TRAVERSING REGULAR POLYGONS’ PATH

In order to explore the mathematics involved in finding the polar equation of regular polygons, we will start with a square. In this calculation, each side of the square is taken to be 2 units and is placed such that the center is on the origin. Each interior angle of a square is  $\frac{\pi}{2}$  rads. As all the sides of the square are straight lines, the first objective is to find one side. It can be easily shown that the side corresponding to the equation  $x = 1$  is  $r = \sec(\theta)$ . In order to get the other three sides of the square, we need to rotate the right side counter-clockwise by subtracting the interior angle,  $\frac{\pi}{2}$  rads from each  $\theta$  in the function and the range. Each rotation creates another side. In order to get a single expression which defines all the sides of the square, we need to use modulo or  $Mod[a, b]$  function. It can then be shown that the polar equation of a square is given by (13).

$$r = \sec \left[ \frac{Mod(\pi + 4\theta, 2\pi) - \pi}{4} \right] \quad (13)$$

A similar procedure can be followed to find the polar equation of n sided regular polygon. The polar equation of an ‘n’ sided regular polygon is given by (14).

$$r = \sec \left[ \frac{Mod(\pi + n\theta, 2\pi) - \pi}{n} \right] \quad (14)$$

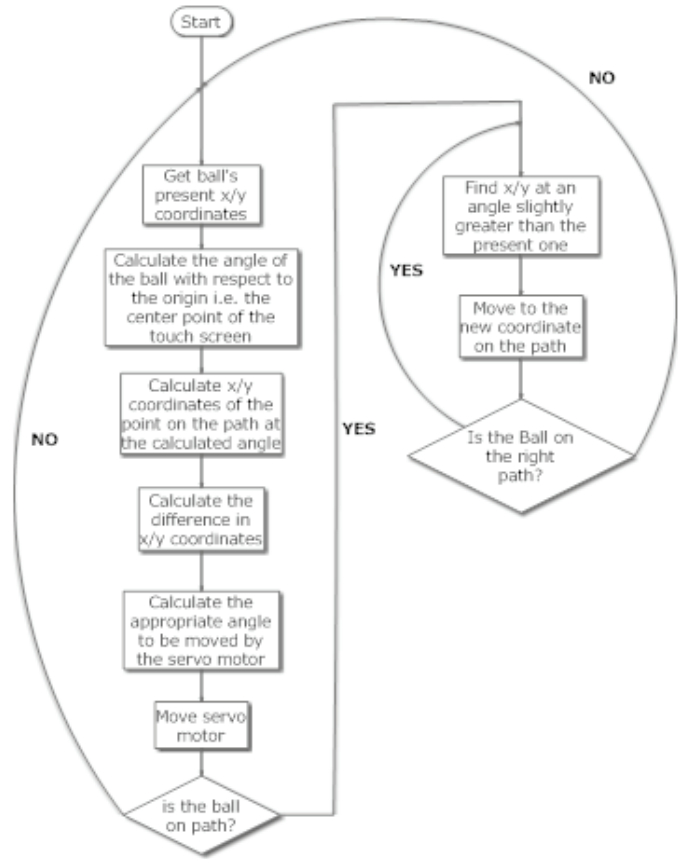


Fig. 8 Flow chart of the polar traversal algorithm

Equation (14), with any integer value of ‘n’, can be fed into the algorithm for the polar graphs to make the ball move on the sides of that polygon. But the method discussed here can only be used for regular polygons. Hence, we now further generalize the discussion to allow an irregular polygon path to be traversed.

## VII. TRAVERSING IRREGULAR POLYGONS’ PATH

It is almost impossible or just too tedious of a task to find one polar equation describing any ‘n’ sided irregular polygon. Hence, the algorithm for the traversal of a general polar path discussed in the previous sections cannot be used here. A different technique was used and a new algorithm was written for the path traversal of the ball on any irregular polygon. Fig. 9 shows 15 sides of an ‘n’ sided polygon. The ball, placed at a random position at the start, has n number of possibilities for moving to any edge of the polygon. Having input all the coordinates of the edges of the polygon, the ball is “aware” of the positions of all the edges. After computing the distance of the ball from each edge, the ball will move to the edge which is nearest to it, which, according to Fig. 9, is 8. Once at 8, the ball will have two possibilities: move to 7 or move to 9. As done previously in traversing polar graphs, a

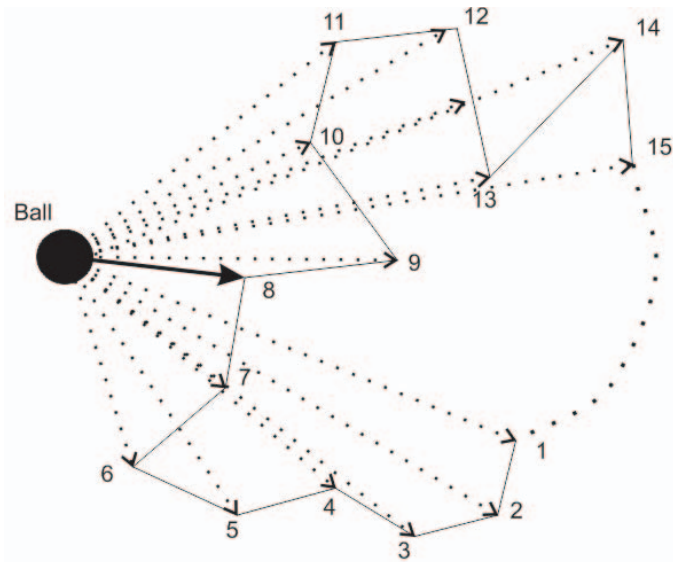


Fig. 9 Irregular Polygon path

direction integer is used to decide whether to move in clockwise or anti-clockwise direction. Let's assume that the direction integer was such that the ball moved to 9 i.e. in clockwise direction. The ball will then move to 10 then 11 then 12 and so on until it has reached the  $n^{\text{th}}$  point. After that, the ball will move to 1 and repeat the path traversed in the previous cycle.

### VIII. CONCLUSION

The work mentioned above was tested on a ball on plate setup and the algorithms worked up to the standards. Using such an approach for the ball on plate system removed the need for the much complex mechanical analysis using Lagrange's equations or Inverse kinematics. To further improve this experiment, a larger and more sensitive touch panel can be used. As a result, the ball can be made to follow the curved paths more accurately as the incremental angle can be made very small. Moreover, this work combined with some image processing in MATLAB can be used to make the ball solve any maze on the touch panel.

### ACKNOWLEDGMENT

The author would like to acknowledge the platform provided by Manufacturing and Automation Research Center at KOC University, Turkey in conducting this research.

### REFERENCES

- [1] H. Wang, Y. Tian, Z. Sui, X. Zhang, and C. Ding, "Tracking control of ball and plate system with a double feedback loop structure," *International Conference on Mechatronics and Automation*, 2007, pp.
- [2] M. Bai, H. Lu, J. Su and Y. Tian, "Motion Control of Ball and Plate System Using Supervisory Fuzzy Controller," *6<sup>th</sup> World Congress on Intelligent control and Automation*, 2006, pp.
- [3] H. Wang, Y. Tian, C. Ding, Q. Gu, and F. Guo, "Output Regulation of the Ball and plate system with a nonlinear velocity observer," *7<sup>th</sup> World Congress on Intelligent control and Automation*, 2008, pp.
- [4] D. Yuan and Z. Zhang, "Modelling and control scheme of the ball-plate trajectory-tracking pneumatic system with a touch screen and a rotary cylinder," *IET Control Theory and Applications*, 2008, pp.
- [5] S. Awtar, C. Bernard, N. Boklund, A. Master, D. Ueda, and K. Craig, "Mechatronic Design of a ball-on-plate balancing system" *Mechatronics*, vol. 1, no. 2, pp. 217-228, Mar. 2002.
- [6] D. Cooper, "Polar Polygons," unpublished.
- [7] M. McRoberts, "Arduino Starter Kit Manual." 2009.