Ball positioning in robotic billiards: a nonprehensile manipulation-based solution

Ball Positioning in Robotic Billiards: A Nonprehensile Manipulation-Based Solution

Senthan Mathavan, *Member, IEEE*, Michael R. Jackson, and Robert M. Parkin, *Member, IEEE*

***Abstract*—The last two decades have seen a number of developments in creating robots to play billiards. The designed robotic systems have successfully incorporated the kinematics required and have had appropriate machine vision elements for a decent gameplay. Independently, computer scientists have also developed the artificial intelligence programs needed for the strategy to play billiards. Despite these developments, the accurate ball manipulation aspect of the game, needed for good performance, has not been addressed enough; two important parameters are the potting accuracy and advantageous cue ball positioning for next shot. In this regard, robotic ball manipulation by predicting the ball trajectories under the action of various dynamic phenomena, such as ball spin, impacts and friction, is the key consideration of this research. By establishing a connection to the methods used in nonprehensile robotic manipulation, a forward model is developed for the rolling, sliding and two distinct types of frictional impacts of billiards balls are developed. High-speed camera based tracking is performed to determine the physical parameters required for the developed dynamic models. To solve the inverse manipulation problem, i.e. the decision on shot parameters, for accurate ball positioning, an optimization based solution is proposed. A simplistic ball manipulator is designed and used to test the theoretical developments. Experimental results show that a 90% potting accuracy and a 100–200 mm post-shot cue ball positioning accuracy has been achieved by the autonomous system within a table area of 6 × 5 ft^2.**

***Index Terms*—Intelligent systems, game-playing robots, nonprehensile manipulation, object tracking, computer vision, manipulators, impact dynamics, intelligent robots, educational robots**

I. Introduction

THE NEED FOR autonomous systems designed to play both strategy-based and physical games comes from the quest to model human behavior under tough and competitive environments that require human skill at its best. In the last two decades, and especially after the 1996 defeat of the world chess champion by a chess-playing computer, physical games have been receiving greater attention. Robocup^{TM}, i.e., robotic football, HuroCup for basketball [1] are well-known examples. The robots created to play the billiards family of games are placed in this context [2]-[3]. The billiards family has a number of variants of which snooker and pool are the most common. The former is quite popular in Europe and has spread to the Far East, hailing TV audiences of hundreds of millions for major tournaments in the last few years [4], while the latter is widely played throughout North America. Although both the games have different rules for pocketing the balls and slightly varied table and ball sizes, from an autonomous game playing robot's point of view, the challenges in terms of ball manipulation, vision, and strategy are entirely similar. Hence, snooker, to play which a robot is considered presently, is frequently referred to as billiards, in a general sense, in this paper. Snooker, as well as being a game of strategy, also requires accurate physical manipulation skills from the player, and these two aspects qualify it as a potential game for autonomous system development research. Although research into playing strategy in billiards has made considerable progress by using various artificial intelligence methods [4, 6], the physical manipulation part of the game, especially the basics of it, is not fully addressed by the robots created so far. This work looks at the different ball manipulation options snooker players use, like the shots that impart spin to the ball in order to accurately position the balls on the table. In this regard, robotic ball manipulation by predicting the ball trajectories under the action of various dynamic phenomena, such as ball spin, impacts and friction, is the key consideration of this paper.

The paper continues as follows. In Section II, a brief review of the systems designed to play billiards is provided. A concise review of non-prehensile manipulation is given in Section III. Section IV introduces the ball manipulation problem in billiards and the necessary theory is set up. Section V treats the specifics of the manipulator system that is built. In Section VI, experimental results are provided. Section VIII reports the results from experiments with the robot manipulator. The paper is concluded with a discussion.

II. A Brief Review of Billiard Robots

In the mid-late 80s, Bristol University developed a robotic snooker player called the Snooker Machine [2]. The robot consists of a PUMA 560 manipulator arm suspended from an SKF Linear Drive System that is mounted as a gantry system above the table. The PUMA arm carries a cue, powered by pneumatics. The vision system consists of an overhead camera to determine the position of balls on the table and another camera mounted on the cue holder to accurately position the cue on the cue ball. Considerable attention has been on developing a playing strategy for the robot. The robot's

S. Mathavan is with Loughborough University, Ashby Road, Loughborough LE11 3TU, UK, ASML, Eindhoven, The Netherlands and Nottingham Trent University, Burton Street, Nottingham NG1 4BU, UK (e-mail: s.mathavan@ieee.org).

M. R. Jackson is with Loughborough University, Ashby Road, Loughborough LE11 3TU, UK (e-mail: m.r.jackson@lboro.ac.uk)

R. M. Parkin is with University of Bradford, Richmond Road, Bradford BD7 1DP, UK (e-mail: r.m.parkin@bradford.ac.uk)

performance is not discussed, except for some special shots, like the ones where the motions of the balls in impact are confined to a straight-line. The strategy subsystem treats various types of impacts by using primitive collision models, like the principle of conservation of linear momentum [7]. In another work, a prismatic, XYZ overhead gantry robot has been used in conjunction with a revolute end effector to play pool [8]. When compared to that of Chang [2], the main difference is on shot selection based on fuzzy-based reasoning [8]. Other works too have had XYZ gantries attached to the tables themselves, e.g. Cheng et al. [9], although such mounting prevents humans and the robots sharing the same table for any competitive gameplay. Another minimalistic, 2 DOF robot consisting of a motorized cue and its yaw control is developed to experiment a certain machine-learning technique called 'reinforcement learning' with it [10].

A major R&D effort to create a pool playing robot, named 'Deep Green', has been going on for the last decade at Queen's University, Canada [11]. An overhead industrial gantry robot carries a suspended robot manipulator, to which a cue stick manipulator is attached (Figure 1). A custom-designed linear manipulator is used to drive the cue stick. The spatial positioning accuracy of the robot is reported to be 0.6 mm and a ball potting accuracy of 67% is claimed for the straight shots [11]. According to their later publication, the robot has pocketed runs of four consecutive balls [3]. However, ball pocketing accuracy and cue ball post-positioning accuracy, which are crucial metrics for a successful and continued gameplay, are not provided. In a recent work an anthropomorphic robot has been developed to play pool [12]. The proposed kinematic solution resembles that of a human. However, for the modeling of billiard dynamics the work resorts to primitive impact models such as that of Wallace and Schroeder [13]. More importantly, the sidespin of balls has not been captured in the impact models used.

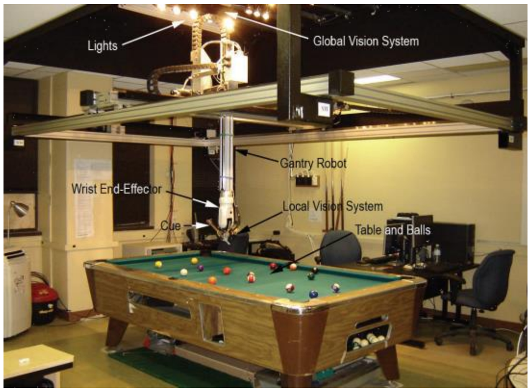


Fig. 1. Gantry-based pool playing system at Queen's University, Canada [2]

The robots configurations found in the literature, be it the gantry based configurations or the humanoid, offer very suitable kinematic solutions for shot executions. The vision subcomponents of the described systems appear to be apt for a billiard robot. The AI part required for the robot has also been tackled extensively [5, 6]. However, a key aspect of the game of billiards is in striking the cue ball by imparting different spins and speeds on it by the cue. The existing billiards systems have handled straight and angled shots whereby the speed and the yaw angle of striking with the cue have been the only parameters controlled with the robot. Whereas, billiard players obtain the required spin and speed combination by choosing the right point on the cue ball to strike. The resulting spin and speed of the cue ball, while potting the object ball, leaves the former at an advantageous spot on the table to maximize the chance of the next strike and so on. This maneuvering is explained in Fig. 2, where depending on the speed and spin imparted, the cue ball ends up in entirely different locations on the table and this ability is vital for any competitive billiard robot of the future.

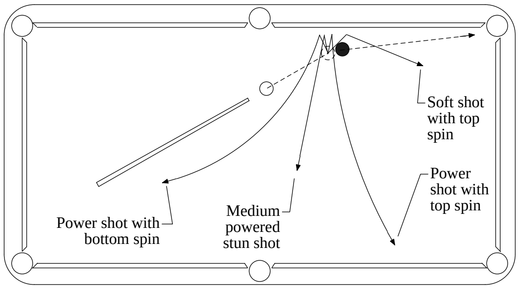


Fig. 2. Final post-shot cue ball locations after a ball-pot for different top and side spins imparted on the cue ball – top view of the table [15]

The reason for the lack of research on the spin manipulation of balls is manifold. The spin dynamics of balls itself has not been discussed thoroughly in the physics literature that addresses spherical bodies. Especially, the impact dynamics of spinning spherical bodies is a special problem that has not been treated comprehensively in other literatures for the roboticists to make use of. From the ball manipulation perspective, a similar situation is found in the nonprehensile robotic manipulation, where the exact dynamic information of the environment is necessary for a robot to manipulate an object without grasping it [16, 17, 18, and 19]. As will be seen later, techniques from robotic nonprehensile manipulation can be applied directly to the current problem. This is the background in which this research work is conducted. Once the manipulation problem is fully developed, the solution can be tested on an existing, full scale gantry-based platform that has the potential to play every billiard shot in a kinematic sense.

III. Robotic nonprehensile manipulation

Nonprehensile manipulation is defined as the manipulation of an object without grasping it (hence also known as graspless manipulation) [16]. Not resorting to grasping the object, a nonprehensile manipulator with fewer degrees of freedom (DOF), uses the gravitational, centrifugal, Coriolis and frictional forces acting on the object as virtual motors to control more DOFs of the object. The object's extra motions are exhibited as rolling, slipping and free flight [20]. Manipulation in a higher dimensional space is essentially achieved by allowing the object to move relative to the robot. Many manipulation methods such as throwing, batting, catching, orienting/reorienting parts, and manipulation by impulses, also come under this category. A clear advantage of such a manipulation scheme is that it does not require very large robots to cover the whole workspace of the object. In snooker, the very nature of the game only allows for nonprehensile manipulation, an impulse-based manipulation to be specific, of the cue ball by the cue. Hence, nonprehensile manipulation methodologies will be extremely useful for a snooker robot.

Nonprehensile manipulation, when compared to grasping manipulation, presents other problems of its own. Grasping manipulation, as in a pick-and-place robot, requires a description of a path for the end effector to follow. This is accomplished by means of programming the individual joints accordingly, and their control is a classical closed-loop control problem. However, in nonprehensile manipulation, once the manipulator releases the object, the object's control thereafter is not with the manipulator. After the object loses contact with the manipulator, its control is taken over by the virtual motors, i.e. the gravitational, frictional or its own inertial effects. From the perspective of the manipulator, the whole scheme is analogous to that of an open loop system, with no feedback control. This 'open loop' nature of nonprehensile manipulation requires a detailed preplanning that considers every aspect of the dynamics of the object motion.

Although there are numerous nonprehensile manipulation methods, as outlined above, only the ones where a force/impact is applied to an object and it is then being allowed to slide on a planar horizontal surface are considered here. In this subcategory of problems, the object, in its 'free' motion, is only under the influence of frictional and inertial forces. This type of problem has been analyzed by many researchers. Impulsive manipulation [21, 22], releasing manipulation [19], tapping manipulation [23, 24], multi-agent dynamic cooperative manipulation [17, 25], pushing manipulation [26] are some examples.

A. Manipulation solution methods used in nonprehensile robots

In nonprehensile manipulation operations, two distinct situations may be encountered: ones where an algebraic analysis of the object dynamics is completely possible and others where this analysis is only partially possible or completely impossible. For positioning flat objects (axi-symmetric and polygonal ones) on a plane with the action of sliding friction, Huang *et al.* [21] and Han and Park [24] use inverse numerical algorithms. Furthermore, there may be situations where, due to the properties inherent to the dynamics of the system, an inverse solution does not exist.

For a given positioning task, instead of finding a direct inverse solution, the manipulation space can also be searched by using its forward dynamics models. This approach is also a solution to situations, where in the absence a direct inverse solution, a reasonable compromise may still be achieved by searching the vicinity of the required object position. For example, a compromise solution may be found, by trying to reduce the error in positioning, using a forward motion model of the object, whilst satisfying any possible constraints on the object motion. Various methodologies have been used in this regard. The major ones employed are nonlinear optimization [16, 17, 18], iterative learning control [19] and machine learning [27].

IV. Theory: Ball Positioning by Nonprehensile Manipulation

For any billiards robot, the desired end position for the cue and object balls are determined by the AI module that makes decisions on the best strategy to win the game in the long run. In this regard, the next object ball to pocket and the best position to leave the cue ball for subsequent shots are decided by the AI module. In this regard, the pocketing of an object ball can also be translated into a positioning requirement. Hence, given these two, final positions required for both balls, initial shot parameters, like cue speed and the impact point on the cue ball (as seen in Fig. 2a) have to be calculated by the robotic shot planner. Given that the ball motion involves different types of dynamics, such as rolling, spinning and impacts in its forward dynamics, the positioning task presented to the robot shot planner is an inverse dynamics problem. As seen earlier in Section III, the knowledge of the dynamics of the manipulated objects, during and after their interaction with the manipulator, is the key for nonprehensile manipulation problems. In this regard, the following subsections describe the manipulation problem and the forward dynamics of ball motion in billiards.

A. Manipulation problem definition

The artificial intelligence (AI) part of the system makes decisions regarding which object ball has to be played next, the pocket into which the object ball must be potted and where to leave the cue ball, after the shot, in order to make the next shot according to the overall game plan. Figure 3 depicts a typical shot scenario in billiards - only a part of the table is shown. Thus, for a given initial cue ball location, $C_{I,}$ the decision to play the ball O_1 into the pocket P_1 and then to leave the cue ball in or very close to the desired ball location C_D has already been taken by the decision-making (AI) sub-system. These results are assumed to be readily available for the robot, as the system described in this paper is devoid of the AI component.

Referring to Figure 3, for a given cue ball location (C_I), targeted object ball (O_1) and pocket (P_1) combination, and to attain a certain desired final cue ball location (C_D), the task is to determine the initial required parameters of the ball motion, namely, linear velocity (V_0), top spin ($\omega^{T}{}_{0}$,), sidespin ($\omega^{S}{}_{0}$) and shot direction (θ), and thereby establish the robot's manipulation parameters: cue speed (V_{C0}), hitting positions on the ball, given by x'_0 and y'_0 in Figure 4(a), and the cue orientation (θ_C). For the proposed robot, θ_C cannot be adjusted automatically, as no swivel DOF is assigned to the robot. Hence, the platform is aligned manually with the feedback

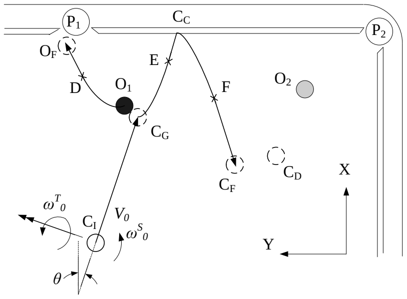
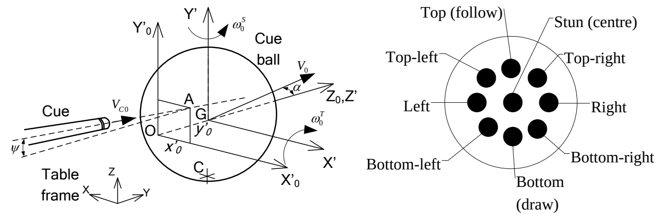


Fig. 3. Typical ball trajectory in snooker

from the overhead camera. For the optimal shot parameter selection, both the shot configuration shown in Figure 3, as well as the most basic shot, where no cue ball – cushion impact takes place, are considered. For a billiards robot, there can be other additional constraints such as other balls very close to the general area of trajectories that limit the possible ball trajectories, and hence the solution space, further. However, the fundamental problem, as defined above, is treated here and the proposed methodology can be extended to include any additional constraints.



(a) (b)

Fig. 4. Geometry in cueing (a), and areas to strike on the cue ball and resulting spins – horizontal view of the player [14] (b)

B. Forward dynamics

1) Cueing

For a given cue manipulation parameters, i.e. cue speed and the position on the cue ball where the cue hits, initial motion parameters such as ball velocity and ball spin must be determined to predict the subsequent motion of different balls on the table. The theoretical cueing model proposed by de la Torre Juarez [28] incorporates all of the effects that are present during cueing. However, the cue tip, being made out of leather or synthetic materials, is highly elastic. Hence, it is very difficult to analytically determine the frictional forces, and their directions, set up between the cue ball and the cue tip. In addition, there are other phenomena such as cue ball squirt [29], where the ball deviates from its ideal movement direction; this deviation angle is denoted by α in Figure 4(a).

For these reasons, it is decided to pursue an experimental approach to determine the dynamics of cueing. For various values of cue speeds V_{C0}, and cue offsets x'_0 and y'_0, the initial ball speed parameters, V_0 and α, are experimentally measured, as described in Section VI.B.2. Then two neural networks are trained with this experimental data to predict V_0 and α for any queried values of cueing parameters, i.e. V_{C0}, x'_0 and y'_0.

In the absence of any reliable experimental data for initial spins of the cue ball, an alternative way has to be found to estimate ω^T_0 and ω^S_0, i.e. top and side spins, respectively. Researchers have often used the assumption of the cue tip gripping the cue ball during their impact [29, 30], largely owing to the fact the cue tip is well chalked before each shot. The tip is also flexible, hence the cue tip is assumed to grip the ball surface as soon the contact is made. A horizontal calibrated camera is used to evaluate typical cue stick inclinations with the horizontal in typical gameplay. From the determined range of about 6° to 20°, the lowest possible angle is chosen as the lower the angle the lower the effects due to the friction between the cue ball and the table [28]. Hence, a cue inclination angle of 6.5° is chosen. For this value, it is shown that the friction force setup between the cue ball and table during cueing is negligible [31]. Now, for an initial cue ball velocity of V_0, the initial ball side and top spins are,

$$\omega_0^S = \frac{5x'_C V_0}{2R^2} \tag{1a}$$

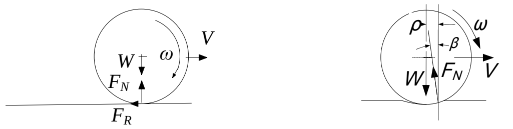
$$\omega_0^T = \frac{5y'_C V_0}{2R^2} \tag{1b}$$

Where, x'_C and y'_C are the initial point of contact between the cue tip and the ball in the X'Y' coordinate system in Figure 4(a) and R is the ball radius. For a given x'_O and y'_O, the values of x'_C and y'_C can estimated from the geometry given in Figure 4(a) and using the cue tip radius of 10 mm. This estimation, in conjunction with Equations (1a) and (1b), leads to the determination of ball spins immediately after cueing.

When the cue inclination ψ is high, i.e. close to 90º, and for high-end cue speeds, a spin about the ball velocity vector V_0 will be imparted to the cue ball (the spin is like the spin of a bullet). This type of shot is known as a massé shot and the frontal spin imparted on to the cue ball will make the ball move along a curved trajectory, making the shot very famous with trick-shot makers in billiards, although professionals hardly use this shot type. For small ψ's, e.g. 6.5° that is used in the current system, the massé effect imparted on the cue ball is negligible.

2) Ball motion on the table

A sphere, such as a billiards ball, that rolls on a very hard surface makes a point-wise contact, as shown in Figure 5(a). However, when either the sphere or the rolling surface is deformable, the contact is not through a point, but via a surface. In billiards, the balls are hard and the table is laid with a soft cloth, resulting in contact scenario shown in Figure 5(b). Moreover, when analyzing the forward ball motion, no consideration is given to the sidespin of the ball as it is assumed not to affect the former. This is known as decoupled motion, where sidespin is considered to be independent of the linear velocity, V or the topspin, ω, of the ball.



(a) (b)

Fig. 5. Ball motion on non-deformable (a) and deformable (b) surfaces

A ball is said to be rolling when,

$$V = R\omega \tag{2}$$

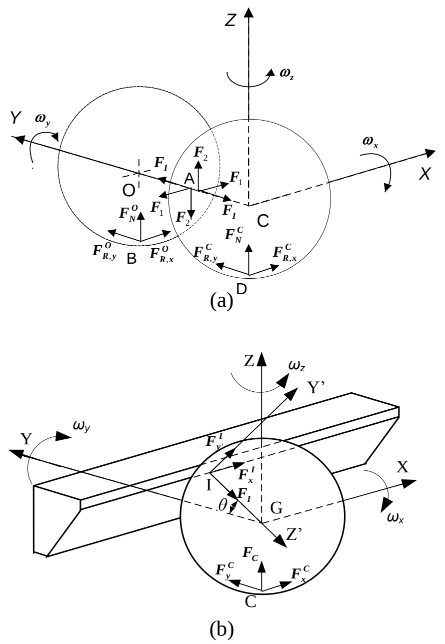
where R is the radius of the snooker balls (Figure 5(a)). If the ball and the surface are non-deformable, as in Figure 5(a), the ball will be in perpetual rolling motion, as the friction forces between the surfaces in contact are zero for the condition in Equation (2). When one of the surfaces is deformable, the ball speed and spin, V and ω, are gradually reduced in the rolling phase. In billiards, the table cloth that makes up the top surface of the table is deformable. Referring to Figure 5(b), the normal force F_N, from the surface does not go through the center of the ball [32]. The normal force F_N produces a horizontal force component and a torque, acting against the linear and rotational motions of the ball. Although the decelerations are not due to friction, an equivalent friction coefficient, μ_r, can be defined for rolling. The linear and rotational decelerations are given by $\mu_r g$ and $\mu_r g/R$, respectively, where g denotes gravitational acceleration.

Rolling usually takes place towards the latter part of ball motion, before it comes to a stop. At the start of motion, the ball slides on the table, in general, and referring to Figure 5, it is either of the following:

$$V > R\varpi \tag{3a}$$

$$V < R\varpi \tag{3b}$$

In the sliding phase, the regular friction coefficient, μ_s, acts on the ball. In snooker and billiards, μ_s is 10 times larger than μ_r [14]. Therefore, when analyzing sliding, one can disregard the effects of rolling. In sliding, the rate of change of linear and angular speeds are given by $\mu_s g$ and $\frac{5\mu_s g}{2R}$. For the condition in Equation (3a), the ball has linear deceleration and angular acceleration. Conversely, when Equation (3b) is true, the ball is under linearly acceleration and rotational deceleration, and this phenomenon is called the 'overspinning'. Given initial conditions, the above expressions for linear and angular accelerations of balls allow the determination of the instantaneous values of the linear ball speed and top (or bottom) spins.



(a)

(b)

Fig. 6. The forces acting on two balls during the impact [33] and the forces setup between a ball and the cushion [15]

The estimation of the sidespin of the ball, at any time instant, is equally important, mainly from the viewpoint of analyzing ball-ball collisions and ball-cushion impacts, where sidespin is a key parameter. As explained later, the table resistance to the sidespin has been measured by tracking the ball using the overhead camera. This resistance acts to retard the sidespin, irrespective of whether the ball has left- or right-spin. The measured resistance provides a means to estimate the spin at any time instant, given the sidespin value at $t = 0$. Referring to Figure 4(b), for balls under right-spin, the change of sidespin ω^Swill be negative, and for left-spinning balls, the change will be positive (the right-hand notation is used here to measure spins).

3) *Collision between two balls*

Traditionally, billiard ball collisions have been analyzed without incorporating the effects of ball spin [13, 28]. The analysis here takes ball spin into account.

For the oblique collision of the balls, as depicted in Figure 6(a), it has been shown in the authors' previous work that, during the duration of impact, either the cue ball or the object ball only stays in contact with the table at any time instant [33]; the other ball is airborne. The determination of which ball remains in contact with the table, at any time during the progression of impact, is entirely decided by the initial ball speeds and spins, at the start of the impact. For the situation where the cue ball is lifted slightly off the table, using Newton's 2nd law of motion for linear and rotational motions of the two balls, the expressions for the simultaneous frictional impacts at points A, B and D, an expression can be derived for the increment of the cue ball velocity in the X direction as,

$$\Delta\dot{x}_G^C = \\ -\frac{\mu_{bb}}{M}\frac{\left(\dot{x}_G^C\right)_n - R\left(\omega_z^C\right)_n - \left(\dot{x}_G^O\right)_n - R\left(\omega_z^O\right)_n}{\sqrt{\left[\left(\dot{x}_G^C\right)_n - R\left(\omega_z^C\right)_n - \left(\dot{x}_G^O\right)_n - R\left(\omega_z^O\right)_n\right]^2 + \left[\left(\dot{z}_G^C\right)_n + R\left(\omega_x^C\right)_n - \left(\dot{z}_G^O\right)_n + R\left(\omega_z^O\right)_n\right]^2}}\Delta P_I \tag{4}$$

Similar expressions can also be written for the other linear and rotational speeds of the cue ball: $\dot{y}_G^C$, $\dot{z}_G^C$, ω_x^C, ω_y^C and ω_z^C (here the superscripts C stand for the cue ball and G is its centroid). In Equation (4), $\mu_{bb}$is the friction coefficient between the two balls, and P_I is the accumulated normal impulse value at A, which is the interface between the balls. M is the mass of the ball and subscript n is the current iteration number denoting the present values for each of the variables with which it is used. Furthermore, six similar equations can also be written for the object ball, O. Equation (4) is essentially a differential equation when $\Delta P_I \to 0$. However, it is not analytically solvable. In order to get ball velocities and spins, Equation (4) and the 11 other equations, for both the balls, are simultaneously solved using numerical integration. P_I is the independent variable, which is increased by small amounts in order to calculate the increments in velocities and spins, as shown in Equation (4). The compression phase of the impact ends when the normal component of the relative velocity between the balls is zero, i.e. along the Y direction,

$$\dot{y}_G^C - \dot{y}_G^O = 0 \tag{5}$$

Here, superscript O stands for the object ball. The accumulated impulse up to the termination of compression, P_I^c, is the value of P_I when Equation (5) is true. The work done by the normal impulse is estimated, for every increment in P_I, by

$$\Delta W_y = (\dot{y}_G^C - \dot{y}_G^O)\Delta P_I \tag{6}$$

Once the compression phase is over, the so-called restitution phase starts, the end of which denotes the termination of impulse. The work done by the normal impulse up to the termination of compression, $W_y(P_I^c)$, is related to the final work done is given by,

$$W_y(P_I^f) = (1-e^2)W_y(P_I^c) \tag{7}$$

Where, P_I^f is the final, accumulated normal impulse value and e is the coefficient of restitution. Once the numerical process passes the condition in Equation (5), in order to determine the termination of impulse, the condition in Equation (7) is checked for after each ΔP_I increment. For this purpose W_y is evaluated by Equation (6), for each iteration, as performed during the compression phase. At the termination of impulse, the application of Equation (4) and the other 11 equations provides the final ball velocities and spins. An extensive description of the algorithm is provided, in full, in the earlier work of the authors [33]. At this point, it shall be worth noting that at the termination of impact both the balls frequently have some frontal spin, as in a massé shot, which makes the balls to curve. The estimation of such curved trajectories is treated in our earlier work in detail [33] and this effect is taken into account in the current trajectory calculations.

4) *Collision between a ball and the side rails*

Similar to the description in the previous section, the collision between a snooker ball and the side rails is a complex dynamic phenomenon, especially when the ball has 3-dimensional spin impact as shown in Figure 6(b). For the first time, the authors of this paper presented a method to analyze such collisions, under the assumption of negligible cushion deformation during the impact [15].

Similar to the analysis of the collision between two balls, using the equations of motion, and equations for the frictional impacts at points I and C in Figure 6(b), the ball's 3 linear velocities and 3 rotational velocities (i.e. spins), with respect to the X, Y and Z axes, can be determined. For the X direction velocity difference between the *n*th and (*n*+1)th iterations of the numerical scheme, as in Equation (4), it can be written that,

$$(\dot{x}_G)_{n+1} - (\dot{x}_G)_n = -\frac{1}{M}\{\mu_w \cos(\Phi)_n + \mu_s \cos(\Phi')_n[\sin\theta + \mu_w \sin(\Phi)_n \cos\theta]\}\Delta P_I \tag{8}$$

Where, $\tan(\Phi)_n = \dfrac{-(\dot{y}_G)_n \sin\theta + (\dot{z}_G)_n \cos\theta + (\omega_x)_n R}{(\dot{x}_G)_n + (\omega_y)_n R\sin\theta - (\omega_z)_n R\cos\theta}$; $\tan(\Phi')_n = \dfrac{\dot{y}_G + \omega_x R}{\dot{x}_G - \omega_y R}$

The superscript is dropped here as only one ball is involved in this collision. The impact at I is the primary impact and the simultaneous impact at C is secondary. In Equation (8), μ_s and μ_w are the friction coefficients between the ball and the table and that between the ball and the side rail, respectively. θ is the angle the common normal between the ball and cushion, at point I, makes with the horizontal. Φ and Φ' are the directions of ball slip at the two impact points, I and C respectively (Figure 6(b)), and these are measured with respect to the X axis. Here too, the equations are not solvable analytically. Once the equations for ball velocities and spins are written a similar procedure to the one described in the previous section is followed to calculate the final velocities and spins. The analysis is extensively treated in the previous work of the authors [15]. One again, spin conditions as in a massé shot can be created at the end of a cushion collision, although before the impact the ball may not have had any frontal spin. This post-impact spin results in a curved ball trajectory and the estimation of which is explained in our earlier work [33]. This phenomenon is taken into account when estimating the trajectories of every ball that collides with a side rail.

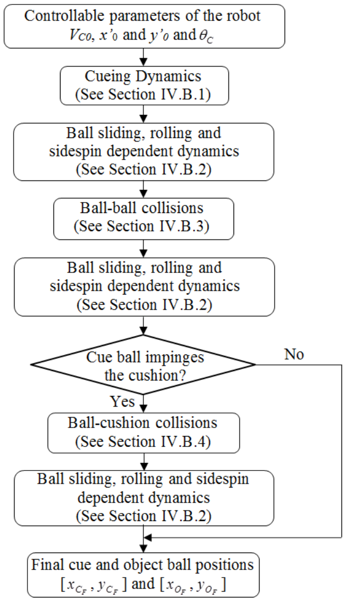

Controllable parameters of the robot
V_{C0}, x'_0 and y'_0 and θ_C
Cueing Dynamics
(See Section IV.B.1)
Ball sliding, rolling and sidespin dependent dynamics
(See Section IV.B.2)
Ball-ball collisions
(See Section IV.B.3)
Ball sliding, rolling and sidespin dependent dynamics
(See Section IV.B.2)
Cue ball impinges the cushion?
No
Yes
Ball-cushion collisions
(See Section IV.B.4)
Ball sliding, rolling and sidespin dependent dynamics
(See Section IV.B.2)
Final cue and object ball positions
$[x_{C_F}, y_{C_F}]$ and $[x_{O_F}, y_{O_F}]$

Fig. 7. Forward dynamics

Now that all possible dynamics encountered in billiards are captured by different models, a typical shot can be analyzed in full. For the ball trajectory shown in Figure 3, the estimation of forward dynamics is as presented in Figure 7.

C. *A solution for the manipulation problem in billiards*

Now, for all the interactions encountered in a typical billiards shot, a form of descriptor for its dynamics is established. The solution for the positioning task, which also includes the potting of the object ball, as set out in Section IV.A, is essentially an inverse one in nature. But some of the descriptors of dynamics are not explicitly expressed by equations, for example, the numerical solutions for the collisions, and the empirical models derived for cueing. This complexity prevents one from obtaining the inverse solutions analytically. Moreover, there may also be positioning requirements for which an inverse solution does not exist. For example, the object ball may not be able to be taken to C_D as required by the AI element, for any combination of manipulation parameters θ_C , x'_0, y'_0 and V_{C0}, so an option would be to take it to a position very close to C_D, for instance to C_F (Figure 3). Hence, searching for a direct inverse solution may not always work. For these reasons, the direct inverse solution based approach is ruled out.

An optimization-based approach is proposed here ball positioning. The optimization function will have to be a composition of spatial errors between the actual positions where the balls will end up for a given shot parameters, and the desired ball locations. The conditions to ensure that the object ball is potted are also a part of the problem. This is generally known as nonlinearly constrained optimization, and can be defined as below [34]. By referring to Figure 3, for $q \in \Re^4$ and also subject to the conditions $V_{C0} < V_{C0,max}$ (i.e. maximum cue velocity possible), $\sqrt{(x'_C)^2 + (y'_C)^2} < 0.5R$ (for no mis-cueing to occur) and $\theta_{min} < \theta < \theta_{max}$ with θ_{min} and θ_{max} ensuring that cue ball C_1 will hit the object ball O_1,

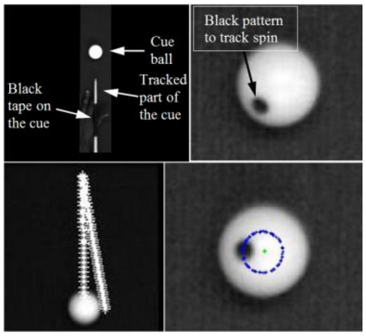


Fig. 8. Tracking features (top row) and two tracked results at 120 and 150 frames per second (bottom row)

$$\text{Minimize } F(q) = \left(x_{C_D} - x_{C_F}\right)^2 + \left(y_{C_D} - y_{C_F}\right)^2,$$
$$\text{subject to } [K(q)] \leq [L], \text{ where, } q = [V_{C0}, x'_0, y'_0, \theta_C] \qquad (9)$$

The matrix condition $[K(q)] \leq [L]$ consists of two elements. This constraint ensures that the object ball is potted by imposing conditions that the trajectory segment O_1O_F should go up to the pocket P_1 (or go past it) and that the minimum distance between the line segment and the center of P1 must be less than half the width of the pocket entrance (for the ball to fall into the pocket). Since $F(q)$ is not differentiable, a derivative-free optimization method must be used. Under similar conditions, a quasi-Newtonian method has been used by Li and Payandeh [17] for planar sliding objects and by Lynch and Black [18] for a batting manipulator. For the present problem, Genetic Algorithms (GAs) are used. The balls' forward dynamics, shown in Figure 7, is coded as a M-function in MATLAB® with q, as defined in Equation (9), as input. The GA-based optimization is performed in MATLAB® Optimization Toolbox.

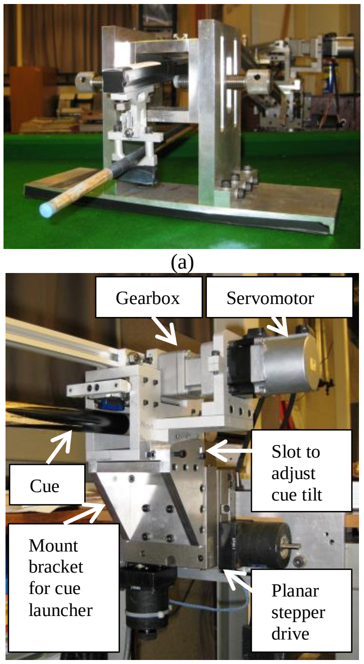
D. *Parameter determination with the high-speed camera*

The forward dynamics model, as explained in the previous subsections, requires a number of physical parameters, such as coefficients of friction. In order to obtain those parameters, tracking experiments are carried out. The white cue ball is used for ball tracking experiments. For interactions involving the cue and the cue ball, a part of the cue, of bright-colored wood, is also tracked. The assumption made in tracking the cue is, it remains close to the horizontal and is at the level of the ball centers. The parameters measured by tracking the cue ball were sliding and rolling coefficients of friction, between the ball and the table, and the coefficients of restitution between two balls and a ball and the side rails. These experiments are described, in detail, in our earlier paper [14]. In addition to these, a circular black pattern is put on the cue ball, as shown on the top-right of Figure 8. The ball is spun sideways about the vertical (i.e. with sidespin only) from a stationary point and is tracked to measure the resistance of the table cloth to sidespin. The rotational deceleration due to cloth resistance is found to be 22 rad/s^2.

V. Cue Manipulator System

A. *Manipulator design*

The material and the geometrical shape of the cue and other features such an elastic cue-tip serve a number of purposes including the suppression of vibration at the time of impact with the cue ball. Hence, a regular, unmodified snooker cue is proposed to be used with the robot. In addition, to allow the regular transverse vibrations of the cue stick at impact, the holding conditions of the cue, which requires a firm grip with the back hand a guiding through the extended front hand, are kept unchanged. The other robots have had either a modified cue or some other mechanical contraption to strike the ball [2, 8, 11]. A rack and pinion system is designed for the linear manipulation of the cue. Based on the requirements, the "200 W Low Inertia System" servo motor, from *SureServo*$^{\text{TM}}$, is selected to drive the pinion via a 3:1 reduction gearbox. A



(a)

(b)

Fig. 9. The designed robotic manipulator, with the passive frontal cue support (a) and the servo and stepper drives with the mechanics (b)

planar (i.e. two-axis) stepper drive, AEROTECH® ATS302, is used so that the cue launcher can be positioned to impart different spins on the cue ball. The mounting detail is seen in Figure 9(b). Depending on the movement effected by the stepper drive the whole cue launcher moves in the vertical plane normal to the cueing direction. There is also a small, constant, tilt of 6.5° introduced in the cue orientation about the horizontal plane.

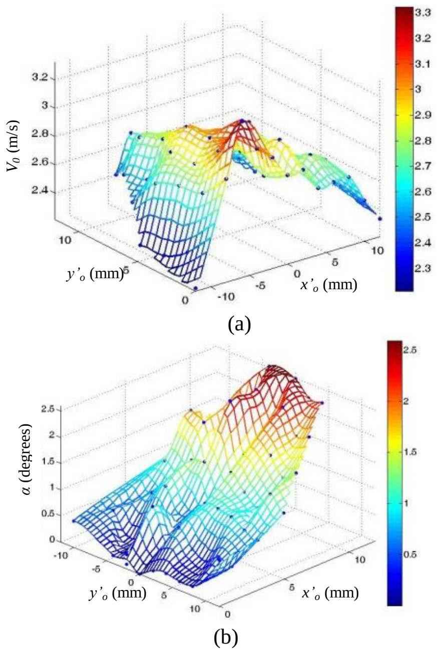
The manipulator is designed with the target of manipulating the cue ball accurately, whereby velocities and spins imparted onto the cue ball can be accurately controlled, unlike in the previous systems. The design does not take into account the kinematics required for an overall billiards robot. It has been indicated earlier, in the literature review, that the gantry-based systems provide an adequate solution to the kinematics problem in billiards. The idea here is to get a solution for the accurate ball manipulation problem separately so that such a manipulator may be incorporated in a gantry based system to achieve the required kinematics. Hence, the designed manipulator cannot reach into the table for all types of shot nor can it move independently around the table, but have to be moved and positioned manually.

B. System description

A Riley Renaissance® professional snooker table of size 10 × 5 ft^2 is used for the experiments described here. A single overhead camera is sufficient to track the balls as trajectories are confined to the plane of the table. Snooker is played with a cue ball in white and 21 object balls of different, but uniform colors, *sans* any patterns. The chosen color camera, PixeLINK PL-B776F, is mounted over the table, pointing vertically down. The camera has a resolution of 3.15 megapixels and contains the region of interest option enabling it to image at up to 1000 frames per second. The camera images a table area of approximately 6 × 5 ft^2 (1.83 x 1.52 m^2) to a 1 mm spatial resolution.

The speed of cueing, thereby the rotational speed of the servo motor, is kept constant throughout the stroke of the cue. The speed of the servomotor, hence the cue speed, is controlled by a PIC18F458 microcontroller board by sending pulses at controlled rates. The pulse rate is set on the microcontroller board from the main program, written in Visual Basic® 6.0 (VB), running on a PC. The vision algorithms are written as M-files in MATLAB®.

When the pulse rate is at its highest, the linear striking velocity of the cue reaches 2.75 m/s, a typical high-end cue velocity found in a normal game of snooker. The rate at which the pulses are sent out from the microcontroller is selected by a string consisting of a 3 digit number sent from the main program to the microcontroller. This 3-digit string ranges from‘001’ to‘200’. String ‘001’ corresponds to a cue velocity of 2.75 m/s and ‘200’ achieves 0.3 m/s. This resolution of the cue velocity can position the cue ball, theoretically, to a 15 mm spatial accuracy on the table, but the repeatability characteristics of the robot, as described later, will also have an effect on the positioning accuracy. The stepper drive, controlling the impact point of the cue on the ball, is driven from the main VB program.



(a)

(b)

Fig. 10. Experimental data for the velocity (a) and squirt(b) of the cue ball against cue impact position for the velocity string ‘070’ (cue speed ~ 2.0 m/s)

VI. Experimental results

The following section describes the experiments performed to test the manipulator system’s performance and the proposed ball positioning strategy.

A. Parameter determination using the manipulator

The white cue ball is always positioned at 75 mm from the cue tip.

1) *Cue ball positioning repeatability*

By recurrently hitting the ball positioned at the same point (a guiding structure is used for this purpose), the repeatability of the robot for shots up to a cue speed of 2.75 m/s is found to be around ±50 mm, on average, for shots involving no cushion impacts and that from a single cushion bounce (snooker ball diameter is 52.4 mm). This deviation is along the direction of travel of the ball, especially for the shots involving no cushion impacts. The positioning repeatability across the main motion direction is found to be ± 10 mm. These repeatability figures are due to the fact that cue tip, made of natural or synthetic leather, is flexible and that chalk is applied to it so that it can grip the smooth cue ball for effective momentum transfer. The chalk is applied after every shot in order to have similar conditions for each shot. This assisted surface friction cannot be uniform between different chalk applications, and combined with the flexibility of the tip results probably in different points on the ball becoming the center of impulse. These impact position changes, although small, must result in variability in momentum transfer, resulting in the aforesaid repeatability figures.

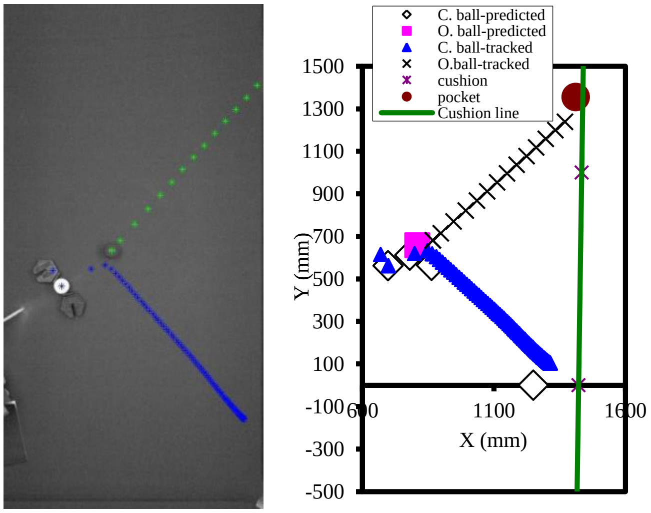
2) *Dynamics of cueing*

The robot manipulator and the cue ball are initially set to have $x'_0 = 0$ and $y'_0 = 0$, where x'_0 and y'_0 are the coordinates of the point at which the line of strike of the cue intersects the X_o’Y_o’ coordinate system, with reference to

Figure 4(a). Five shots of different cue speeds at approximately V_{C0} = 0.5 m/s intervals (in the range of 0.5 m/s to 2.5 m/s) are executed for the same ball position, by replacing the ball back to the initial position after each shot. y'_0 is varied from -12 mm to 12 mm in increments of 2 mm, by using the stepper drive, and for each y'_0, x'_0 is varied from zero to 12 mm also in 2 mm increments. Negative values of x'_0 are not tested as the impulse geometry is symmetric about OY$_0$' in Figure 4(a). For each combination of x'_0 and y'_0, 5 shots are played at different cue speeds, as mentioned earlier. For each shot, resulting ball velocities and cue ball squirt are measured from by ball tracking with the camera. The data obtained, for a given cue speed, for both initial ball parameters, are plotted in Figures 10(a) and 10(b) respectively. This experimental data is used to train a back propagation feed-forward neural network (NN) to predict cue ball speeds for any given values of shot parameters, i.e., x'_0, y'_0 and V_{C0}. A 3-5-1 neuron configuration is found to be suitable. A similar NN is trained to predict ball squirt as well. These two NNs serve as the cueing model, as explained in Section IV.B.1.

B. *Ball manipulation tests with the robot*

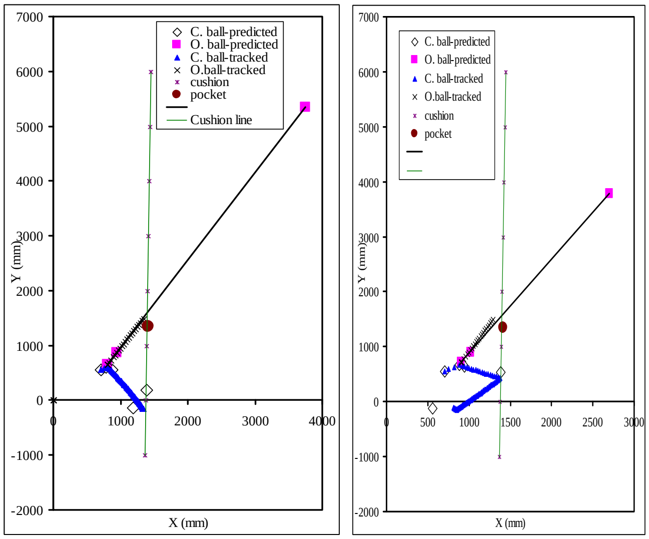
Using the camera, the pocket locations on the part of the table that is in the field of view of the camera, and the cushion (i.e. side rail) lines are established. The initial cue ball location is determined using the camera. A red ball is used as the object ball and its position on the table is established by processing the R component of the color image sequences. The platform is moved around the table manually to hit a ball within the 6 × 5 ft^2 area. Once this is done the position of the cue is established, in relation to the table and pockets, by imaging from the overhead camera. Only the shots for which the cue holding backhand is outside of the table area can be executed with the manipulator as it is attached to the rigid platform. The shots where the player reaches into the table cannot be played. The two initial ball positions are embedded in the M-function describing the forward dynamics. The desired final cue ball location is also specified. The M-file is then called from the Optimization Toolbox using its function handle and executed to deliver the best value for q.



(a) (b)

Fig. 11. Positioning results

Experiments on ball positioning are performed within a table area of 6 × 5 ft^2 (the full size, professional table is 10 × 5 ft^2). For the following values of θ_C = 0.515 rad, $[x_C, y_C]$ = [698 mm, 562 mm], $[x_O, y_O]$ = [869 mm, 681 mm], and a desired cue ball location of $[x_{C_D}, y_{C_D}]$ = [1250 mm, 0 mm], the optimization routine has predicted the following parameters for the robot: V_{C0} corresponding to string '073', x'_0 = -11 mm and y'_0 = 0 (x'_0 and y'_0 are approximated to the nearest millimeter) without any cue ball-cushion collision. Hence, the shot imparts left spin onto the cue ball; the top spin is zero. The shot that is executed for the above parameters is shown in Figure 11(a) with the predicted trajectory (the pocket is not seen in the figure as it is right next to the right side edge of the image). The ball is potted and the cue ball, without any collision with the cushion, is positioned at 110 mm from its desired location. Moreover, it is worth noting that it is explained in IV.B.3 and IV.B.4 that after an impact the balls will have frontal spins, in general, resulting in curved ball trajectories. However, the amount of frontal axis spins that result from the impact do not lead to noticeable deviations, making the predicted trajectories being near straight lines. With reference to Figures 11 and 12, this fact is also confirmed in the experimental results where the camera system does not measure any deviations from straight line trajectories.



(a) (b)

Fig. 12. More positioning results showing a shot that is potted (a) and a missed shot

Figure 12(a) shows another shot where the object ball is potted, and the cue ball is positioned with 83 mm error after the shot. The shot parameters are: V_{C0} = '100', x'_0 = 8 mm, y'_0 = -6 mm, imparting a bottom-right spins combination to the ball (also see Figure 2(b)). Here, the algorithm predicted a ball-cushion collision, whereas the cue ball has not hit the rail, but ended up near touching the cushion. An instance of the worst shot performed with the manipulator is shown in Figure 12(b), where the object ball has missed the pocket and the cue ball was placed at 248 mm from the predicted position; the shot parameters are: V_{C0}: ‘100’ , x'_0 = 0 mm, y'_0 = -10 mm (bottom spin).

Within this area of the table, for the shots that the manipulator could make with its kinematics, an object ball potting accuracy of more than 90% is obtained. In addition, the cue ball is positioned to accuracy within the range of 100-250mm, in general. The Neural network based prediction for the cueing model works almost in real-time with MATLAB®. The numerical impact models take 5 seconds to run. The GA based optimization takes 3-4 minutes to complete, and is the bottleneck.

VII. Discussion

As robots are made both for pool and snooker, the main variants in the billiard family of games, a comparison between the relative difficulties of automated gameplay must be performed to benchmark the different systems proposed for the two games. In this regard, some facts concerning the pocket and the ball sizes in pool and snooker must also be considered. In snooker, all six pockets are 90 mm in size and the ball diameter is 52.5 mm. If the mid-pocket entry is considered to be ideal for an object ball in snooker, the margin of maximum allowable error for a flawless entry (not touching the pockets) is around 19 mm, on either side of the ideal line of entry. However, the way the cushion near the pocket entrance is shaped allows up to a 45 mm deviation for the corner pockets and a 55mm for the middle pockets, in snooker. Pool balls are 52.5 mm in diameter. In pool, the four corner pockets are 114-117 mm in size while the middle pockets measure 127-130 mm [35]. This leaves a robot with the margin of error of 28.5 mm for the corner pockets and 35 mm for the middle pockets, for a non-contact entry of the object ball; thus, the maximum allowable distance offset values can also be expected to be larger than those in snooker. The preceding comparison underlines the fact that the ball potting is difficult in snooker. Hence, if the same robot is employed to play both the games, it will have a higher potting accuracy in pool when compared to snooker.

The performance of the current robot must be evaluated in light of other facts concerning the robot and the forward dynamics model for the ball motion. The robot’s repeatability in ball positioning is found to be around ±50 mm and this, in turn, will affect the positioning accuracy of the robot. There have been vibration problems with the robot and the servo drive system is proposed to be replaced with a linear servo actuator [31]. This is possibly one reason for the potting issues, such as the one presented with the shot in Figure 12(b). In addition, a very basic model is utilized for the spin estimates of the cue ball immediately after cueing, neglecting the friction from the table. Hence, the forward model itself is not perfect and more accurate spin tracking with a higher resolution camera will establish a better model for cueing dynamics. The optimization problem can possibly be reformulated giving lesser weight to the solutions where multiple impacts that may give rise to additional uncertainties will be present. For example, a forward solution with only a cue ball-object ball impact should be given an advantage over a solution where the cue ball makes an additional impact with the cushion before positioning itself to the desired location. To bring the optimization timing down, other methods such as the Nelder-Mead method may be used, although in snooker there is no maximum stipulated time (referee can always intervene) with professional players taking up to a minute to think through the table state and make a shot.

VIII. Conclusion

This paper presented a novel work on using a minimalistic manipulator to play the game of billiards like highly skilled players. Professional players make use of top and side spins of the cue ball, in addition to the accurate controlling of its speed, to position balls accurately on the table. In this regard, the paper presents models for different types of dynamics involved in ball motion. Making use of similarities between the present manipulation problem and non-prehensile manipulation research, where a considerable amount work has been done, a ball manipulation strategy is established. By designing a manipulator that can not only control the cue speed tightly but also the impact point on the ball - thereby ball spins - the ball potting-positioning problem in billiards is tested. The manipulator is rigidly mounted to a platform that sits next to the table. Moreover, the platform is manually moved around the table and the cue cannot reach all areas in the billiards table used. Tests showed that the manipulator could pocket the object ball 90% of the time within a table area of 6 × 5 ft^2. A post-shot cue ball positioning accuracy of 100-200 mm is also achieved within the same area. The manipulations methodologies and models used in the work could be experimented on one of the full-fledged gantry robots used for billiards gameplay toward a future system that could come close to beating humans at this adept task.

References

[1] C. H. Kuo, T. S. Chen, H. C. Chou and G. Z. Chen, "PN-WSNA-based Eye-hand-leg Coordination with a FIRA HuroCup Basketball Game," *IEEE/ ASME Trans. on Mechatronics*, vol. 18, no. 3, pp. 854-866, 2013.

[2] S. W. S. Chang, “Automating Skills Using a Robot Snooker Player,” Ph.D. dissertation, University of Bristol, Bristol, UK, 1994.

[3] M. Greenspan, J. Lam, M., Godard, I. Zaidi, and S. Jordan, “Toward a Competitive Pool-Playing Robot”, *Comput.* vol. 41, no. 1, pp. 46-53, 2008.

[4] World Snooker 2014, Available: http://www.worldsnooker.com/page/NewsArticles/0,,13165~2352434,00.html

[5] J.-P. Dussault, and J.-F. Landry, “AI Optimization of a Billiard Player”, *J. of Intell. and Robotic Syst.*, vol. 50, no. 4, pp. 399-417, 2007.

[6] M. Smith, “PickPocket: A computer billiards shark”, *Artificial Intell.*, vol. 171, no. 16-17, pp. 1069-1091, 2007.

[7] K. H. L. Ho, T. Martin, and J. Baldwin, “Snooker Robot Player - 20 Years On”, in *IEEE Symp. on Computational Intell. and Games (CIG 2007)*, Honolulu, Hawaii, 2007, pp: 1-8.

[8] M. E. Alian and S.B. Shouraki, “A Fuzzy Pool Player Robot with Learning Ability”, *WSEAS Trans. on Electronics*, vol. 2, no. 1, pp. 422–425, 2004.

[9] B. -R. Cheng, J. -T. Li, and J.-S. Yang, “Design of the Neural-Fuzzy Compensator for a Billiard Robot”, in *IEEE Int. Conference on Networking, Sensing & Control*, Taipei, Taiwan, 2004, pp. 909-913.

[10] A. W. Moore, D. J. Hill, and M. P. Johnson, An Empirical Investigation of Brute Force to choose Features, Smoothers and Function Approximators, in S. Hanson, S. Judd, and T. Petsche, editors, Computational Learning Theory and Natural Learning Systems, Volume 3. MIT Press, 1994.

[11] F. Long, J. Herland, M. -C. Tessier, D. Naulls, A. Roth, G. Roth, and M. Greenspan, “Robotic Pool: An Experiment in Automatic Potting”, in

IEEE/RSJ Int. Conference on Intell. Robots and Syst., Sendai, Japan, pp. 2520-2525, 2004.

[12] T. Nierhoff, K. Heunisch, and S. Hirche, “Strategic play for a pool-playing robot”, in *IEEE Workshop on Advanced Robotics and its Social Impacts*, Munich, Germany, pp. 72-78, 2012.

[13] R. E. Wallace and M. C. Schroeder, “Analysis of Billiard Ball Collisions in Two Dimensions”, *Amer. J. of Physics*, vol. 56, no. 9, pp. 815-819, 1988.

[14] S. Mathavan, M. R. Jackson, and R. M. Parkin, “Application of high-speed imaging to determine the dynamics of billiards”, *Am. J. Physics*, vol. 77, no. 9, 788–794, 2009.

[15] S. Mathavan. M. R. Jackson, and R. M. Parkin, “A theoretical analysis of billiard ball dynamics under cushion impacts”, *Proc. of the IMechE, Part C, J. of Mech. Eng. Sci.,* vol. 224, no. 9, pp. 1863-1873, 2010.

[16] M. T. Mason, “Progress in Nonprehensile Manipulation”, *Int. J. of Robotics Research*, vol. 18, no. 11, pp. 1129-1141, 1999.

[17] Q. Li and S. Payandeh, “Planning Velocities of Free Sliding Objects as a Free Boundary Value Problem”, *Int. J. of Robotics Research* , vol. 23, no. 1, pp. 69-87, 2003.

[18] K. M. Lynch and C. K. Black, “Recurrence, Controllability, and Stabilization of Juggling” *IEEE Trans. on Robotics and Automation*, vol. 17, no. 2, pp. 113-124, 2001.

[19] C. Zhu, Y. Aiyama, T. Arai, and A. Kawamura, “Positioning in Releasing Manipulation by Iterative Learning Control”, *J. of Int. and Robotic Syst.*, vol. 46, no. 4, pp. 383-404, 2006.

[20] K. M. Lynch, and M. T. Mason, Dynamic Nonprehensile Manipulation: Controllability, Planning, and Experiments. *The Int. J. of Robotics Research* 18(1), pp: 64-92. ISSN: 0278-3649, 1999.

[21] W. H. Huang, E. P. Krotkov, and M. T. Mason, “Impulsive Manipulation”, In *IEEE Int. Conference on Robotics and Automation*, Nagoya, Japan, pp. 120-125, 1995.

[22] C. B. Partridge, and M. W. Spong, Control of Planar Rigid Body Sliding with Impacts and Friction. *The Int. J. of Robotics Research* 19(4), pp: 336-348. ISSN: 0278-3649, 2000.

[23] W. H. Huang, and M. T. Mason, Mechanics, Planning and Control for Tapping. *The Int. J. of Robotics Research* 19(10), pp: 883-894. ISSN: 0278-3649, 2000.

[24] I. Han and S. –U. Park, “Impulsive Motion Planning for Positioning and Orienting a Polygonal Part”, *Int. J. of Robotics Research*, vol. 20, no. 3 , pp. 249-262, 2001.

[25] Q. Li, and S. Payandeh, Multi-Agent Cooperative Manipulation with Uncertainty: A Neural Net-Based Game Theoretic Approach. *Robotics and Automation, 2003. Proc. ICRA '03. IEEE Int. Conference on*, 14-19 Sept 2003 Taipei, Taiwan. IEEE. pp: 3607-3612. ISBN: 0-7803-7736-2.

[26] N. Rezzoug, and P. Gorce, Dynamic Control of Pushing Operations. *Robotica* 17(6), pp: 613-620. ISSN: 0263-5747, 1999.

[27] M. Matsushima, T. Hashimoto, M. Takeuchi, and F. Miyazaki, “A Learning Approach to Robotic Table Tennis”, *IEEE Trans. on Robotics*, vol. 21, no. 4, pp: 767-771, 2005.

[28] M. de la Torre Juarez, M, “The Effect of Impulsive Forces on a System with Friction: The Example of the Billiard Game”, *European J. of Physics* , vol. 15, no. 4, pp. 184-190, 1994.

[29] R. Cross, “Cue and ball deflection (or “squirt”) in billiards”, *Am. J. of Physics*, vol. 76, no. 3, pp. 205-212, 2008.

[30] D. G. Alciatore,”Fundamentals”, in *The Illustrated Principles of Pool and Billiards*, Sterling Publishing, New York.

[31] S. Mathavan, “Trajectory Solutions for a Game-Playing Robot Using Nonprehensile Manipulation Methods and Machine Vision”, Ph.D. dissertation, Loughborough University, Loughborough, UK, 2009.

[32] J. Hierrezuelo and C. Carnero, “Sliding and Rolling: The Physics of a Rolling Ball”, *Physics Educ.*, vol. 30, no. 3, pp. 177-182, 1995.

[33] S. Mathavan. M. R. Jackson, and R. M. Parkin, “Numerical simulations of the frictional collisions of solid balls on a rough surface”, *J. of Sports Eng.*, vol. 30, no. 3, pp. 227-237,2014.

[34] J.-S. R. Jang, C. -T. Sun, and E. Mizutani, ”Derivative-based optimization”, in *Neuro-Fuzzy and Soft Computing - a Computational Approach to Learning and Machi. Intell*., 1st Ed., Prentice Hall , New Jersey.

[35] World Pool-Billiard Association 2014, Available: http://www.wpa-pool.com/

S. Mathavan (M’10) received the BSc degree in Mechanical Engineering from the University of Peradeniya, Sri Lanka, in 2005. He then obtained the PhD degree in Mechatronics from Loughborough University in the UK, in 2009. He held a Rolls-Royce funded post-doctoral research position in Intelligent Automation at Loughborough University from 2010 to 2011.

Since 2011, he works as a mechatronics engineer at ASML in the Netherlands and he is also a visiting researcher at Loughborough and Nottingham Trent Universities in the UK.

M. R. Jackson received the B.Sc. (Hons.) degree in Mechanical Engineering from Salford University, Salford, UK, in 1975, the M.Sc. degree in Flexible Manufacturing Systems and Robotics from Loughborough University, in 1987, and the Ph.D. degree in machine tool vibration and surface finish from the Council for National Academic Awards (CNAA) and Leicester Polytechnic, Leicester, UK, in 1986.

He is currently Professor of Machine Systems and Director of the EPSRC National Centre for Innovative Manufacturing in Intelligent Automation at the Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University.

R. M. Parkin (M’94) received the B.Sc. degree in Engineering Science from Leicester University, Leicester, UK, and the Ph.D. degree in diesel gas turbines from Leicester Polytechnic, Leicester, in 1976 and 1980, respectively.

He was Professor of Mechatronics at Loughborough University, where he was also the Head of the Wolfson School of Mechanical and Manufacturing Engineering. Since 2014, he is Pro-Vice-Chancellor (Research and Knowledge Transfer) at University of Bradford, UK. Prof. Parkin was also a founder member of the UK Mechatronics Forum.