1. being able to move away from the active table space to not interfere with the balls

Post-shot Ball Interference

In order to adequately strike a ball anywhere on the table the end-effector or striker will be required to enter the active ball area (the theoretical volume at which object inside it may interfere with a rolling ball). As a result the end-effector must enter this volume to initiate contact with the ball but must not stay inside it much longer to allow the balls to roam freely. Whether it’s a linear or rotational actuator the active motion of exiting the volume must occur.

Our team will address this issue by creating a mechanism that will raise the end-effector soon after the shot. This will be done as quickly as possible without damaging the instruments or requiring human intervention to reset it to its striking position. According to current discussions the team is considering a raising motion although we have yet to finalize any designs so the outcome may change if other (and better) solutions arise.

1. Counting steps to know the current location and orientation of the end-effector

Counting Steps

When the system is required to move from one location to another it will be required to account for the number of steps it needs to move to get there. This is why being able to count the number of steps of all motion motors (linear motor systems for translational motion and rotational motor systems for system orientation) is essential to the success of the project. A difficulty which we may encounter during this project is both the real time tracking of steps and the loss of steps due to disturbances or system error.

We will address this issue in several ways. Firstly, the system must be well built in order to not allow slack for the motors to slip or skip steps. Second, we are estimating that most if not all motors will be stepper motors which have a definite mechanical step rather than servos which rely on other tracking methods. We will have a dedicated system within the microcontroller software that will track each step carefully in order to avoid inaccuracies. If the system detects a potential miscalculation there will be known checkpoints that the system will use to calibrate the stepping sequence. These checkpoints are not necessarily often since they are not meant to replace the stepping mechanism of the motor but they should be frequent enough to the program to reset often. The actual number of checkpoints is based on the accuracy of the motor and structure of the system (i.e. an inaccurate system will require lots more checkpoints rather than a very exact solution). I predict somewhere between 2 and 10 checkpoints depending on the integrity of the motion.

3) avoiding recoil from the motion of striking the ball

The end-effector will be required to strike at many levels of force in order to score from different positions on the table. Some of the more powerful shots may induce large amounts of recoil in the robotic arm. In order to avoid interfering with the other balls on the table and to avoid damaging the system, the recoil from a shot must be mitigated.

To effectively mitigate the recoil of the end-effector, a real-time control system must be implemented. The control system will invoke a force opposite to the recoil of the system in order to maintain the stability of the system.

4) controlling the force at which the end-effector strikes so that it can score from different positions on the table

Since the system will be required to make shots from most positions on the billiards table, the end-effector must strike at a variable force. After the system takes an image of the table, and processes it through the VR software, it will calculate a vector for the end-effector to strike.

Once the system calculates the vector, it will send the distance value to the arduino. Using the distance value, the arduino will then calculate the optimal PWM value to send to the actuator that controls the end-effector's striking motion.

1. being able to reach all locations on the table (are there any shots we wont be able to reach?

Pre-shot Ball Interference

In order to strike the cue ball according to specifications (no spins or trick shots) the end effector needs to be at the same level than the cue ball to hit it at the center. This arises dificulties due to the fact that some other balls may be at that same position or near enough so that the end effector wouldn't fit without moving it. Also, the rails consist of another fixed physical obstacle that would interfere in the same way.

This problem will be tackled by software. The system will take the position of all the balls and the spacing available to place the end effector. Using this information, the system will only shoose a shot that doesn't have any interference.

There could be the case when there wouldn't be any space to place the end effector in striking position due to lack of space. In this cases the system will make a concesion from the pool rules and pass on its turn, giving the human player the chance to play again. A specific signal will be used to let the human player know that such event has occured either by the use of an led or a sound.

1. providing power supplies to all the necessary components

Power Supplies

Due to the characteristics of the system, different components have different voltage and current requirements, therefore different power supplies need to used of each of this components. The PC come with its own power supply and therefore another one is not needed. For the microcontroller (arduino mega) it requires a 7-15 V DC power supply, we will use a transformer that connects to regular power outlet and outputs a 12 V DC. In the microcontroller's case current is not a limiting factor since it requires relativelly low current compared with the 5 amp output of the transformer. The other components that need its own power supply are the actuators. For this case we will use another power supply depending on the power requirements of the actuators. We will use a controller for the actuators in order to protect the microcontroller. In the case that we use NEMA 17 stepper motors then a good choise is A4988 Stepper Motor Driver For Arduino. In the case that we other higher power steppers or different actuators then other controllers will be used. There is many power supplies available as well as actuator controllers so we are confident that will be able to find one that meets our needs.

1. Figure out if motors will be able to move the robot. Make estimate of weight and see which motor specs do we need.

Motor Specs

Due to the nature of the task, the end-effector is required to have a high range of motion while also being highly precise. The implication of these requirements is that the mechanical build of the robot arm will need to be rather extensive, and in effect, could be of considerable weight. Given that the mechanical build could be rather heavy, it is pertinent to consider motor specs required to move the machine.

The most pressing concern is specs required to facilitate movement along first dimension of motion (the length of the table most likely) as this will require being able to move the entire weight of the machine. For our purposes we will use a liberal estimate of 250 kg for the mass of the robot. We will likely be using some form of slide bearing to aid movement of the machine along the first dimension of motion. We will therefore use an estimate of 0.3 for coefficient of static friction (mu). This is, again, a very liberal estimate as coefficient of static friction for slide bearings can vary anywhere from approximately 0.003 to 0.3. Using the mass of the robot and the coefficient of static friction, we find that a force of 250\*9.81\*0.3 = 735.75 N is required to move the machine along the first dimension of motion.

While 735.75 N is a rather high value, it is not entirely unfeasible. It is possible to purchase stepper motors rated for 50 N-m stall torque. This translates to 735.75 N of force at roughly a 6.8 cm effective radius (radius to translate motor rotation to linear motion). As it is likely that we will be using an even smaller radius for this purpose, 6.8 cm is a perfectly acceptable metric. If for some reason a 50 N-m motor is found to be unfeasible it is possible to use multiple motors of lesser capability, two 25 N-m motors for example. Furthermore, intelligent planning with regards to the design of the robot arm could result in a total weight much less than 250kg and the use of high quality slide bearings could result in a coefficient of static friction orders of magnitude less than 0.3. Both of these factors would work to reduce the required stall torque of the motors.

1. Adequate emergency shut-off controls.

Emergency Shut-off

Given that the robot arm of the system is likely to be quite heavy, the motors used to drive the machine will be required to be rated for high torque. High torque components such as these pose a serious safety hazard as loose material may become trapped in the moving parts and may be unable to be pulled free. It is therefore important that an individual operating the mechanism or otherwise interacting with the system should be able to kill power to the machine at any time during which high torque components may pose a threat to that individual.

For this requirement to be met, a viable solution is to design the system such that accessible emergency shutoff controls be located on all moving parts which operate with high torque. In this way, if at any given time an individual operating the system is to become trapped by the moving components, an emergency shutoff control is guaranteed to be within reach. Furthermore, it would be prudent to place backup emergency shutoff controls at regular intervals along the edge of the pool table. The intention being that for all possible configurations of the robot arm at least one primary shutoff control on the mechanism and one backup shutoff control on the table be within reach if an individual operating the system is to become trapped by the moving components.

9) Rigid structure to avoid bending or unwanted motion

Rigid Structure

A large portion of the system will include rails which will provide support for translational motion for the end-effector. As a result there must be non-bending bars that will be able to span the long distances of the table. Furthermore all jitter, shaking or unnecessary motion must be reduced in order to provide stability for the system and avoid inaccuracies in the motion.

This is will be addressed through the selection of non-flexing materials. These materials will be sturdy and straight at the length required by the design. Additionally, it is planned to apply support points along the way which will ensure straightness if the bar is deemed too long. Finally, the system must be designed to include very little jitter space within connections. All mechanical connections must be tight and be able to withstand disturbances from the environment.