

Quantum Galton Board Implementation: Summary

Hayden Dyke
Imperial College London
London, United Kingdom
hayden.dyke23@imperial.ac.uk

Ismail Mehmood
Imperial College London
London, United Kingdom
ismail.mehmood24@imperial.ac.uk

Abstract—The Galton Board, also known as a Bean Machine, is a model of the Plinko Game, where a ball is modelled as dropping through a series of pegs, with a given probability of falling left or right at each. We here present a summary of the implementation of a quantum version of the classical Galton Board, inspired heavily by [1].

I. INTRODUCTION

We shall present a summary of our understanding of the implementation of a quantum Galton Board. We shall outline first the workings of the classical Galton Board to serve as a base of intuition, before then moving to an unbiased quantum version, and finally to a biased quantum version (the Universal Statistical Simulator [1]).

II. CLASSICAL GALTON BOARD

A model of the Plinko Game, the classical Galton Board arranges pegs in a triangular form, with balls dropped from the top. At each peg, the ball has a 50/50 chance of falling to the right or left. Fig. 1 shows a typical representation.

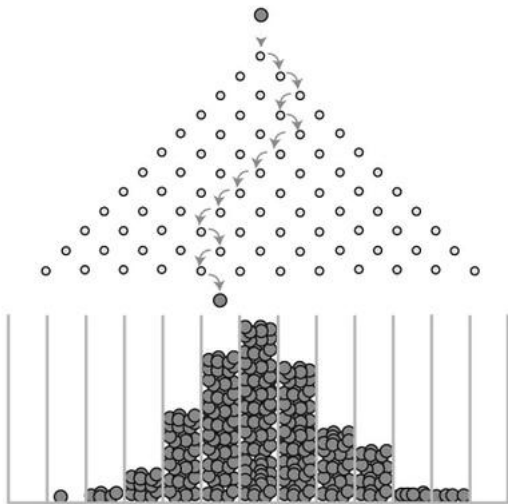


Fig. 1: A representation of a classical Galton Board. The resulting distribution of balls is binomial, with $p = 0.5$. [2]

For n levels of pegs, it is apparent that there are $n + 1$ possible exit columns. Each of these is denoted by $k = \{0, \dots, n\}$ from left to right. If we assign a rightward move a value of $+1$, and a leftward move a value of -1 , then the

final position can be given a value X , where, for a given column k ,

$$X = k(1) + (n - k)(-1) = 2k - n \quad (1)$$

and the probability of achieving a given score is represented by,

$$P(X = x) = \binom{n}{k} p^k (1 - p)^{n-k} \quad (2)$$

evidently a binomial distribution (with $p = \frac{1}{2}$ in this case). The De Moivre-Laplace Theorem, itself a special case of the Central Limit Theorem, states that as n increases, this approaches a normal distribution, motivating the results we expect from an unbiased Galton Board.

III. UNBIASED QUANTUM GALTON BOARD

A Quantum Galton Board is simply a quantum circuit that implements identical logic to the classical Galton Board, using quantum bits (qubits) instead of classical bits. In the unbiased case, we therefore wish to achieve the same Normal distribution of results as in the classical case. We shall use the framework from [1] to explain the quantum circuit implementation of such a board.

A. A single unbiased peg

Following the modular approach of [1], we first discuss the implementation of a single unbiased quantum peg. We make use of 4 qubits, labelled q_0, q_1, q_2, q_3 , where q_0 is the control qubit, q_2 represents the input qubit (i.e. this is where the ball enters), and q_3, q_1 represent the output of the ball along left or right trajectories respectively. To represent the ball entering the peg, we initialise all four qubits to $|0\rangle$, and then apply an X gate to q_2 , resulting in a state of $|q_3 q_2 q_1 q_0\rangle = |0100\rangle$. To then implement the unbiased peg logic, we apply a Hadamard gate to the control qubit, which places it into a balanced superposition, modelling the equal probability of falling left or right. We utilise a controlled-SWAP to swap the states of q_1 and q_2 depending on the control qubit, before utilising a controlled-NOT gate to enforce a state of $|1\rangle$ for the control qubit. A final CSWAP then controls the fall (if any) to the left. We measure q_1 to determine a rightwards fall, and q_3 to assess a leftwards fall. The final resulting state of the circuit is given by,

$$\frac{1}{\sqrt{2}}(|1001\rangle + |0011\rangle) \quad (3)$$

which represents an equal probability of falling left (with $q_3 = 1$) or right (with $q_1 = 1$). The implementation of this circuit in [1] is shown in Fig. 2.

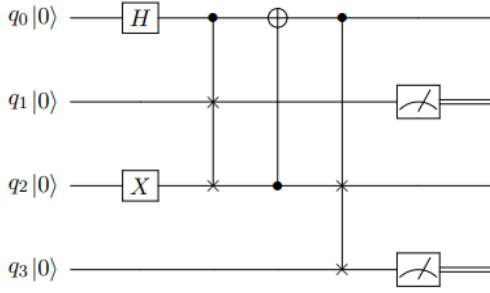


Fig. 2: The single-peg circuit described above, from [1].

B. Extending to multiple levels

Thanks to this modular approach, the extension of this to n levels is largely straightforward, with a few extra steps each time. For example, with $n = 2$, we have 3 pegs, and so we need $n + 1 = 3$ output channels and $n + 1 = 3$ ancillae, one of which is the control qubit. After each level, we reset q_0 to $|0\rangle$ and then reapply the Hadamard gate to q_0 to re-implement the balanced superposition. We can then simply repeat our one-peg CSWAP-CNOT-CSWAP structure twice, once for each peg on level 2. The only difference is that we need to apply a CNOT between the logic of each peg, which ensures the control qubit is in a balanced superposition throughout. Measurements for our three possible outcomes are then taken from q_5, q_3, q_1 respectively. It is trivial to see that this process generalises for n levels using the modular structure in [1], by resetting and re-superposing the control qubit between layers, and utilising the CSWAP-CNOT-CSWAP peg logic once for each peg in the layer. Each peg's logic separated by a CNOT to enforce the equal superposition, which ensures that each peg remains unbiased. For $n = 2$, the circuit is shown in Fig. 3.

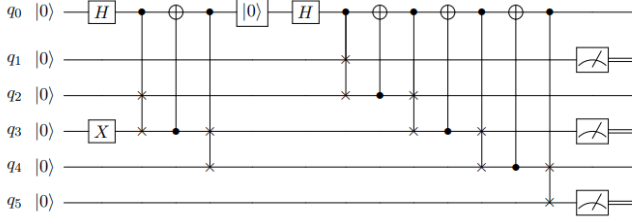


Fig. 3: The circuit for a three-peg quantum Galton Board, from [1]. This generalises to n levels as described.

As [1] confirms, the resulting local simulated distribution of outcomes is indeed a normal distribution as n increases, as expected, though the noise associated with the real quantum hardware simulation should be noted.

IV. BIASED QUANTUM GALTON BOARD

We now turn to the biased version, which effectively allows each peg to have different probabilities of a left or right fall. This can be used to precipitate any statistical distribution, hence the term ‘Universal Statistical Simulator’ [1].

A. A single biased peg

The implementation of the biased peg is similar to the unbiased peg, but with a key difference. As opposed to the Hadamard gate, we apply a controlled rotation about the x -axis of the Bloch sphere, represented by,

$$R_x(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -i \sin(\frac{\theta}{2}) \\ -i \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \quad (4)$$

Using $\theta = \frac{\pi}{2}$ would thus allow us to recover the Hadamard gate, but in general, this produces a biased fall. This results in the single biased peg circuit in Fig. 4.

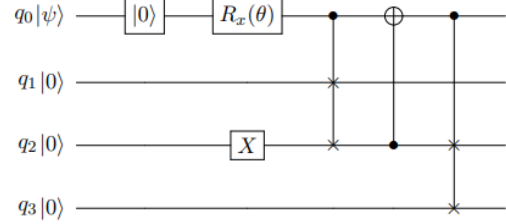


Fig. 4: The circuit for implementing a single biased peg, from [1].

As [1] demonstrates, the use of $\theta = \frac{2\pi}{3}$ produces a final state of,

$$\left(\sqrt{\frac{3}{4}}|0011\rangle + \frac{1}{\sqrt{4}}|1001\rangle \right) \quad (5)$$

which gives a 75% probability of a rightwards fall, with only 25% for a leftward fall. By varying θ , we also vary this probability ratio, allowing us to achieve any desired split.

B. Extending to multiple levels

We can then extend this approach to multiple levels, achieving fine-grained control over the specific bias for each peg. We still reset the control qubit to $|0\rangle$ between layers as before, but now, we need to apply a $|0\rangle$ reset and a custom controlled-rotation before each individual peg too, which ensures we achieve the desired bias for each peg. We need also to reset the control qubit to $|0\rangle$ at the end of each layer, ready for the next layer. Finally, to replace the CNOT correction from before, if a row has i pegs, we need $i - 1$ CNOTs to correct outputs at the end of each layer. For $n = 3$, we thus arrive at the circuit in Fig. 5 (from [1]), which has removed the final reset for clarity.

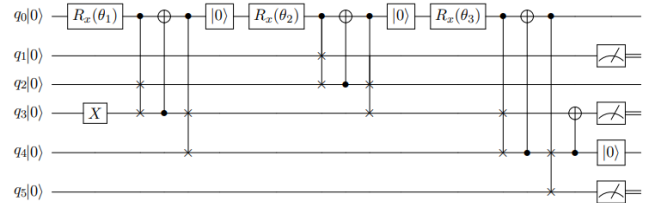


Fig. 5: The circuit for a three-peg biased quantum Galton Board, with individual control over the bias of each peg, from [1]. This generalises to n levels as described.

REFERENCES

- [1] M. Carney and B. Varcoe, “Universal Statistical Simulator,” 2022, [Online]. Available: <https://arxiv.org/abs/2202.01735>
- [2] P. Elston, “Randomness and the Galton Board,” 2022, [Online]. Available: <https://www.chimpinvestor.com/post/randomness-and-the-galton-board>