

# Improvements to the Digital Curling System

KENTARO MORI<sup>†1</sup> TOSHIYUKI HIGUCHI<sup>†1</sup> TAIKESHI ITO<sup>†1</sup>

**Abstract:** “Digital Curling” is a system for analyzing curling strategies. Curling AIs are developed over time and problems of the system are illuminated. In this study, we reconstruct and improve the system to make it more flexible. We will hold a tournament with a different rule called “Mix Doubles” with the improved system.

**Keywords:** Digital curling, Physical Simulator, Server Client System

## 1. Introduction

The “Digital Curling” system was made for analyzing the pure strategy of curling [1][2]. The system simulates curling matches through 2D physics simulations. A human player or AI player (called a “Curling AI”) can play matches against other humans or Curling AIs. Digital Curling tournaments are held several times a year [3].

These tournaments have seen significant success and competitive Curling AI programs have improved over time. However, certain issues were illuminated during such tournaments based on player feedback (e.g., artificiality of physics models of stones, difficulty of rule modification, and poor protocols for communication with Curling AIs).

In this study, we resolve these issues and improve system flexibility for further rule modification. We re-construct the Digital Curling system and propose a “mix doubles” Digital Curling server.

## 2. Digital Curling System

### 2.1 Overview

The Digital Curling system enables competitive play between two Curling AI programs on a local computer. This system consists of two binaries: the physics simulator and game server (fig.1).

The system combines uncertain elements (e.g., condition of the ice, skill of players, etc.) into a random number, which is then applied to the shot vector. The system does not consider the “sweep” element.

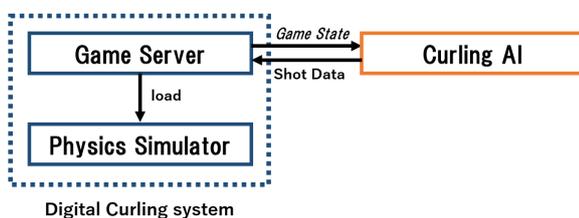


Figure 1 System Configuration

### 2.2 Physics simulation

A physical simulator called “CurlingSimulator” is distributed as a dynamic link library (DLL). This simulator calculates the motion of stones and state transitions.

The physical mechanism for the “curl” (curve) of stones

remains unexplained. To imitate the curl, the simulator applies a constant vertical force to the velocity vector at each step of the simulation.

The simulator library provides functions called “Simulation,” “CreateShot,” and “CreateHitShot.”

#### Simulation

The Simulation function simulates a shot. Given the current state of a match (coordinates of stones, number of shots, number of ends, scores on previous end) and a shot vector with initial velocity, the server simulates the shot and updates the match state.

#### CreateShot

The CreateShot function calculates a shot vector from the coordinate where the stone will stop. The vector is calculated with a heuristic expression.

#### CreateHitShot

The CreateHitShot function calculates a shot vector from the coordinate where the stone will pass through and the “weight” (strength of the shot).

These functions can be called from programs that load the CurlingSimulator.dll

### 2.3 Game Server

A game server program called “Digital Curling – Client” is distributed through the web [4]. The server controls the state of a match and communication between Curling AIs. The server first loads the simulator. When a shot vector is sent from Curling AIs, the server simulates the shot utilizing the simulator.

The server and Curling AIs exchange messages through commands. These commands are defined as a protocol and published through the web [4].

### 2.4 Issues

#### 2.4.1 Physics model

- **Random number model**

Current systems utilize a rectangular coordinate system (fig.1). A pair of random numbers is added to  $(x, y)$  components of the shot vector independently. However, real curling players determine the “weight” and direction of a shot. Therefore, a pair of random numbers should be added to a  $(v, \theta)$  coordinate in a polar coordinate system (fig.2) rather than a rectangular coordinate system.

<sup>†1</sup> The University of Electro-Communications

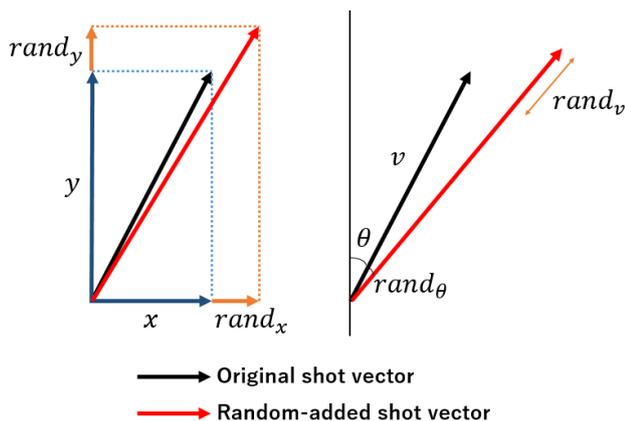


Figure 2 Rectangular (left) and polar (right) random number model

• **Curl of stone**

The simulator imitates a curl by applying a constant vertical force to the shot vectors. Curling players have indicated that the curl effect is weak in current systems. By increasing the constant value, the intensity of the curl will be increased.

However, the CreateShot function operates based on a heuristic expression, meaning that we must modify the function to able to change the constant value.

• **Flexibility for rule modifications**

When a rule is changed, we should modify the source code and publish a rebuilt binary. However, there are two binaries with different random numbers within the same application version, which often leads to confusion during tournaments. The system should be more flexible in order to facilitate rule modification without replacing the binary. The rules and regulations of the tournament will be changed in the near future.

**2.4.2 Protocol**

The protocol provides certain commands for communication between the server and Curling AIs. However, there are some issues with these commands (e.g., there is no command to concede a match for Curling AIs, and there are few commands to send information regarding a match). For modifying rules, the server and Curling AIs should exchange more information.

**3. Improvements to the System**

**3.1 Overview**

We reconstructed the current system while resolving the issues discussed in section 2. The simulation procedure is basically same as the current system. Variables in the simulator that can be changed based on rules can be modified by the server. In the improved simulator, these variables are defined as non-constant values and have interfaces for external modification. The server can load user options and apply them to the simulator without replacing the binary.

Furthermore, we implemented a random number

generator with polar coordinate system and server that can operate the network utilizing an extended protocol.

**3.2 Physics simulator**

**3.2.1 Random number model**

A random number generator with polar coordinate system was added to the improved simulator. The server can choose which generator (polar or rectangular) to use. The server can also set the size of each component of the random numbers.

**3.2.2 Flexibility for rule modifications**

As mentioned in section 3.1, the server can change variables in the simulator. The improved simulator has a class called “Simulator” (fig.2), which contains non-constant member variables. When the server simulates a shot, it creates an instance of the simulator and sets its values according to the rules and regulations. Therefore, rule modifications can be applied easily without replacing the binary.

“NumberOfFreeguard” and “AreaOfFreeguard” variables are parameters of the “Free Guard Zone” (FGZ) rule. The FGZ rule restricts the first *n* stones in an end to remove in-play stones in the FGZ. The number of stones (*n*) and area of the FGZ are different between basic games and “Mix Doubles” games. “RandomType” represents which generator to use (mentioned in 3.2.1). These variables are public and can be changed after an instance is created.

“StoneDensity” represents the density between stones and the ground (ice). “ShotTable” stores the shot data for generating shots (mentioned in 3.2.3). This table is initialized when the instance is created. “StoneDensity” affects the trajectory of the delivered stone; hence, these two variables are defined as private members and cannot be changed.

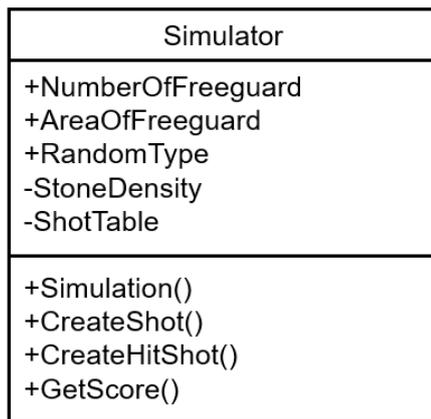


Figure 3 Simulator class

**3.2.3 Shot generator**

We reconstruct the “CreateShot” and “CreateHitShot” functions. These functions should be flexible for modification of the “StoneDensity” (fig.3) or other values which affect the trajectory of a stone.

When the instance of Simulator is created, the simulator runs the single simulation for recording the coordinates and velocity

vector at each step to tables. A shot vector is generated utilizing the tables.

### 3.3 Game server

#### 3.3.1 Protocols

We extend and redefined the protocol as “Digital Curling Protocol” (DCP). We added three commands: “CONCEDE,” “GAMEINFO,” and “SETSTONE.”

The CONCEDE command allows Curling AIs to concede (surrender) a match. Curling AIs can send this command instead of the “BESTSHOT” command, which sends a shot vector. After receiving this command, the game server notifies both Curling AIs that the game is over.

The GAMEINFO command is sent from the server to provide players with information regarding the match (size of random numbers, type of random number generator, type of rules, etc.).

The SETSTONE command is sent from a Curling AI to place a stone in mix doubles curling. This command has an arrangement 0, 1, 2, 3, which represents the pattern of the stones (discussed in section 5).

#### 3.3.2 Network server

The current system runs only on the Windows OS and can simulate matches only on a local PC. In previous Digital Curling tournaments, two Curling AIs have run on a single PC arranged by the tournament organizer. In these tournaments, Curling AI developers had to construct binaries that could run on the tournament PCs, which restricted Curling AI development for tournaments.

Therefore, we should develop a tournament environment for networked tournaments. For this reason, the improved server will support TCP/IP for communication to Curling AIs.

## 4. Mix Doubles Curling [5]

### 4.1 Rules

In mix doubles curling, each team delivers five stones in an end and competes for the highest scores in eight ends. A team consists of two players, one male and one female. One player delivers the first and fifth shots while the other delivers the second, third, and fourth shots in each end.

Before an end starts, two stones are placed at the center line such that one is in the house and the other is in front of the house (fig.4). The stones can optionally be placed on the side of the play area (fig.5). This option is called a “power play.” Each team can make one power play per match. The team that loses (did not score) the last end chooses these options. At the first end, the referee chooses the team by flipping a coin. The place of stone A in figs.4 and 5 depends on the amount of curl. The place is never changed during a game [6].

The first three shots cannot remove any stones from the play area. If a delivery violation occurs, the delivered stone will be removed and all stones will be replaced as they were before the shot.

Figure No. 1 - Centre Guard

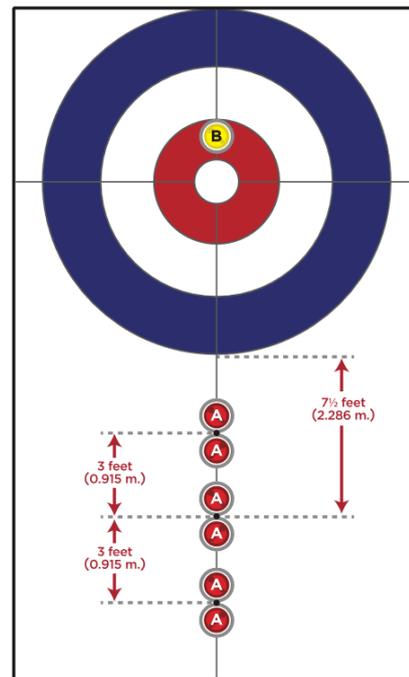


Figure 4 Placement of stones [5]

Figure No. 2 - Power Play Option

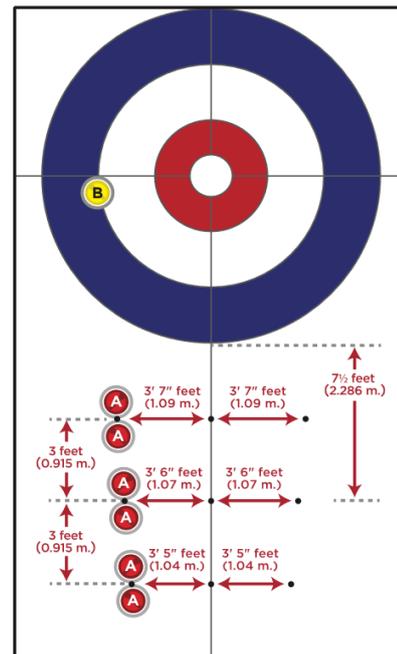


Figure 5 Placement of stones (Power Play Option) [5]

### 4.2 Application to Digital Curling

We applied the mix doubles curling to the improved system. The server notifies which rules will be applied with the GAMEINFO command. Before each end starts, the server sends the SETSTONE command to either player. The player is chosen randomly at the first end.

The place of stone A is fixed in the middle of the options shown in figs.4 and 5, which is 2.286 m below the bottom edge of the house and 1.07 m left from the center line in the “Power Play Option.”

## 5. Further Work

We have implemented the improved simulator discussed in section 3.2 and the server with extended protocol discussed in subsection 3.3.1. This system can simulate a mix doubles curling game with Curling AIs on a local PC. The source code for these programs and the protocol for the server are published on the web [4].

In future work, we will hold a tournament of mix doubles curling for Curling AIs. Furthermore, we will implement the network server discussed in subsection 3.3.2. Based on the tournament results and player feedback, we will evaluate the effects of our improvements.

We consider modifying more complex rules (changing density as time proceeds, sweeps, etc.) in the future.

## References

- [1] Takeshi Ito and Yuuma Kitasei, *Proposal and Implementation of "Digital Curling,"* 2015 IEEE Conference on Computational Intelligence and Games (CIG2015), pp. 469–473 (2015)
- [2] Yuuma Kitasei, Raita Okada, and Takeshi Ito, *Introduction of the proposed Digital curling server,* Information Processing Society of Japan, the 31<sup>st</sup> meeting of the Game Informatics Research Group, GI-31(2) (2014)
- [3] Takeshi Ito and Kentaro Mori, *Report on Digital Curling Tournament in 2015 fiscal year,* Entertainment Computing 2016, 2016-EC-41, pp. 1–6 (2016)
- [4] *Digital Curling,* from <http://minerva.cs.uec.ac.jp/curling/wiki.cgi> (retrieved 04-25-2018)
- [5] *Rules and Regulations, World Curling Federation,* from <http://www.worldcurling.org/rules-and-regulations-downloads> (retrieved 04-25-2018)
- [6] Sean Turriff, *Curling: Steps to Success,* ISBN: 978-1-4925-1577-7, pgs.161~162 (2016)

## Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 15H02797.