# Robot Baseball: A Stochastic Markov Game Model

Antón Seijo Barrio

## Abstract

This document summarizes the mathematical formulation and computational approach used to solve the *Robot Baseball* problem developed by the Artificial Automaton Athletics Association (Quad-A). Each at-bat is modeled as a stochastic, zero-sum Markov game between a pitcher and a batter. The objective is to determine the probability of reaching a full count (3 balls, 2 strikes) under optimal mixed strategies for both players and to find the parameter $p$ (home run probability) that maximizes this probability.

## 1 Game Description

At the beginning of each at-bat, the count is $(b, s) = (0, 0)$. At each pitch:

- The **pitcher** chooses to throw either a *ball* or a *strike*.
- The **batter** chooses to either *wait* or *swing*.

The outcomes are:

| Pitcher | Batter | Result |
|---|---|---|
| Ball | Wait | $b \to b + 1$ |
| Strike | Wait | $s \to s + 1$ |
| Ball | Swing | $s \to s + 1$ |
| Strike | Swing | Home run with prob. $p$; otherwise $s \to s + 1$ |

The at-bat terminates if:

- $b = 4$: walk, batter scores 1 point.
- $s = 3$: strikeout, 0 points.
- Home run occurs: 4 points.

## 2 Mathematical Model

The game is modeled as a finite Markov decision process with states represented by the pair

$$(b, s), \quad \text{where } b \in \{0, 1, 2, 3\}, \quad s \in \{0, 1, 2\}.$$

These represent the current number of balls and strikes, respectively.

The terminal (absorbing) conditions are:

$$V(4, s) = 1 \quad \text{(walk, 4 balls)},$$
$$V(b, 3) = 0 \quad \text{(strikeout, 3 strikes)}.$$

Thus, there are $4 \times 3 = 12$ non-terminal states plus the terminal outcomes.

## Bellman Equation

For non-terminal states:

Summing the expected values of all outcomes gives:

$$\begin{aligned} V(b, s) = &(1-x)(1-y)\,A \\ &+ \big[(1-x)y + x(1-y) + xy(1-p)\big]\,B \\ &+ xy\,p \cdot 4. \end{aligned}$$

where

$$A := V(b+1, s), \qquad B := V(b, s+1),$$

and $x, y \in [0, 1]$ are the mixed strategies:

Simplyfing, we obtain

$$V(b, s) = A + (B - A)(x + y - xy) + p(4 - B)xy,$$

- $x$: probability the pitcher throws a strike.
- $y$: probability the batter swings.

## Mixed Strategy Equilibrium

In a mixed strategy equilibrium, each player randomizes their actions so that the opponent becomes **indifferent** between their own options.

**Idea.** The pitcher chooses the probability $x$ of throwing a strike such that the batter's expected value of *waiting* equals that of *swinging*. Similarly, the batter chooses the probability $y$ of swinging so that the pitcher's expected loss (or the batter's gain) is the same whether the pitcher throws a ball or a strike.

**Mathematically.** Let the expected value to the batter when the pitcher throws a ball ($x = 0$) be

$$V_B = (1 - y)A + yB = A + (B - A)y,$$

and when the pitcher throws a strike ($x = 1$):

$$V_S = (1 - y)B + y\,[p \cdot 4 + (1 - p)B] = B + y\,p(4 - B).$$

At equilibrium, the pitcher must make the batter indifferent between these two outcomes:

$$V_B = V_S.$$

This yields:

$$A + (B - A)y = B + y\,p(4 - B).$$

Solving for $y$ gives the batter's optimal mixed strategy:

$$\boxed{y^* = \frac{A - B}{p(4 - B) + (A - B)}}.$$

By the symmetry of the previous equation, the pitcher's equilibrium mixing probability satisfies the same expression:

$$\boxed{x^* = y^*}.$$

**Interpretation.** This condition ensures that neither player can improve their expected payoff by deviating to a pure strategy. If the resulting $x^*$ or $y^*$ falls outside the interval $[0,1]$, the equilibrium collapses to a **pure strategy** (i.e., one player always chooses the same action).

### Boundary Conditions

$$V(4,s) = 1, \quad V(b,3) = 0.$$

## 3 Probability of Reaching a Full Count

Let $r(b,s)$ denote the probability of reaching state $(3,2)$ before termination.

### Recursion

$$r(b,s) = (1-x^*)(1-y^*)\, r(b+1,s) + [(1-x^*)y^* + x^*(1-y^*) + x^*y^*(1-p)]\, r(b,s+1).$$

Boundary conditions:

$$r(3,2) = 1, \qquad r(4,s) = 0, \qquad r(b,3) = 0.$$

Note that there is no additional term for the home run outcome, since a home run immediately ends the at-bat and thus cannot lead to the target state $(3,2)$.

Thus $q(p) = r(0,0)$ is the probability that an at-bat reaches a full count given parameter $p$.

## 4 Optimization Procedure

1. For a given $p \in [0,1]$, solve $V(b,s)$, $x^*$, $y^*$ by backward induction.
2. Compute $q(p) = r(0,0)$.
3. Search numerically for
$$p^* = \arg\max_p q(p).$$

## 5 Numerical Implementation

A Python implementation uses dynamic programming and a grid search over $p$. The key functions are:

- `solve_for_p(p)`
- `find_optimal_q()`

## 6 Results

The numerical search yields:

$$p^* \approx 0.227, \qquad q_{\max} \approx 0.296.$$

Hence, under optimal play and the best choice of home run probability, approximately 29.6% of at-bats reach a full count.

# 7 Repository Structure

```
robot-baseball/

 README.tex           # This document
 README.md            # Markdown version (optional)
 robot_baseball.py    # Python implementation
 q_of_p_table.csv     # Computed q(p) values
 analysis.ipynb       # Interactive notebook
```

## License