```javascript
/**
 * Generate an odd-order magic square using the Siamese method.
 * @param {number} n - Size of the magic square (must be odd and >= 3)
 * @returns {number[][]} Magic square matrix
 */
function generateMagicSquare(n) {
    if (!Number.isInteger(n) || n < 3 || n % 2 === 0) {
        throw new Error("Size must be an odd integer >= 3");
    }

    // Initialize empty n×n matrix
    const magicSquare = Array.from({ length: n }, () => Array(n).fill(0));

    let num = 1;
    let i = 0; // start row
    let j = Math.floor(n / 2); // start column

    while (num <= n * n) {
        magicSquare[i][j] = num;

        // Move up and right
        let newI = (i - 1 + n) % n;
        let newJ = (j + 1) % n;

        if (magicSquare[newI][newJ] !== 0) {
            // If cell is already filled, move down instead
            i = (i + 1) % n;
        } else {
            i = newI;
            j = newJ;
        }
        num++;
    }

    return magicSquare;
}

/**
 * Verify if a matrix is a magic square
 * @param {number[][]} matrix
 * @returns {boolean}
 */
function isMagicSquare(matrix) {
    const n = matrix.length;
    const magicSum = (n * (n * n + 1)) / 2;

    // Check rows
    for (let row of matrix) {
        if (row.reduce((a, b) => a + b, 0) !== magicSum) return false;
    }

    // Check columns
    for (let col = 0; col < n; col++) {
        let sum = 0;
        for (let row = 0; row < n; row++) sum += matrix[row][col];
        if (sum !== magicSum) return false;
    }
```

```
    // Check diagonals
    let diag1 = 0, diag2 = 0;
    for (let i = 0; i < n; i++) {
        diag1 += matrix[i][i];
        diag2 += matrix[i][n - i - 1];
    }
    if (diag1 !== magicSum || diag2 !== magicSum) return false;

    return true;
}

// Example usage:
try {
    const n = 5; // odd number
    const magicSquare = generateMagicSquare(n);

    console.log(`Magic Square of size ${n}:`);
    magicSquare.forEach(row => console.log(row.join("\t")));

    console.log("Is magic square valid?", isMagicSquare(magicSquare));
} catch (err) {
    console.error("Error:", err.message);
}
```

How it works:

1. **Siamese Method**:
    - Start in the middle of the top row.
    - Place numbers sequentially, moving **up and right**.
    - Wrap around edges.
    - If the target cell is already filled, move **down** instead.
2. **Magic Constant**:

$$\text{Magic Sum} = \frac{n(n^2+1)}{2}$$

3. **Validation**: Checks all rows, columns, and both diagonals.

Example Output for `n = 5`:

```
Copy code
17      24      1       8       15
23      5       7       14      16
4       6       13      20      22
10      12      19      21      3
11      18      25      2       9
Is magic square valid? true
```