# Hexapod Control Comparison Test Methodology

## Overview

This document outlines a comprehensive test methodology for comparing two control approaches in hexapod locomotion:

- **Inverse Kinematics (IK)** - Traditional mathematical approach
- **Neural Circuit** - Spiking neuron-based control

The comparison will be conducted across three distinct terrain environments:

1. Obstacle-filled terrain
2. Rough terrain (Perlin noise-generated)
3. Ramp terrain (variable steepness platforms)

---

## 1. Performance Metrics

### Primary Metrics

| Metric | Description | Units | Measurement Method |
|---|---|---|---|
| **Navigation Success Rate** | Percentage of successful task completions | % | Binary success/failure per trial |
| **Completion Time** | Time to traverse/complete terrain challenge | seconds | Start to finish timing |
| **Energy Efficiency** | Total joint torques/forces over time | N·m·s | Sum of absolute joint forces |
| **Stability Index** | Measure of balance and fall incidents | score | Custom stability algorithm |

### Secondary Metrics

| Metric | Description | Units | Measurement Method |
|---|---|---|---|
| **Adaptability** | Response time to terrain changes | seconds | Time to adjust after disturbance |
| **Precision** | Accuracy in reaching target positions | meters | Distance from target coordinates |
| **Gait Consistency** | Variability in step patterns | coefficient | Standard deviation of step timing |

---

## 2. Test Scenarios by Terrain Type

### A. Obstacle Terrain Tests

#### Scenario 1: Navigation Challenge

- **Objective**: Navigate from start to end point avoiding obstacles
- **Configurations**:
  - Low density: 5 obstacles
  - Medium density: 10 obstacles
  - High density: 15 obstacles
- **Success Criteria**: Reach target without collision
- **Environment Parameters**:
  - Area: 28.5m × 29m
  - `minSpacing`: 0.3m
  - `alturaExtra`: 0.1m

#### Scenario 2: Precision Maneuvering

- **Objective**: Navigate through narrow passages
- **Configurations**:
  - Wide passages: `minSpacing = 0.8m`
  - Medium passages: `minSpacing = 0.5m`
  - Narrow passages: `minSpacing = 0.3m`
- **Success Criteria**: Pass through without contact
- **Measurement**: Contact detection via collision events

#### Scenario 3: Dynamic Adaptation

- **Objective**: Navigate while obstacles are randomly repositioned
- **Configuration**: Obstacles repositioned every 10 seconds
- **Success Criteria**: Adapt path in real-time without stopping
- **Duration**: 2 minutes continuous navigation

### B. Rough Terrain Tests

#### Scenario 1: Roughness Progression

- **Objective**: Traverse increasing terrain difficulty
- **Configurations**:

| Level | Roughness Scale | Amplitude | Difficulty |
|---|---|---|---|
| 1 | 0.5 | 0.3 | Easy |
| 2 | 1.0 | 0.6 | Medium |
| 3 | 1.5 | 1.0 | Hard |
| 4 | 2.0 | 1.5 | Extreme |

- **Success Criteria**: Maintain forward progress without falling
- **Distance**: 10m traverse per level

#### Scenario 2: Endurance Test

- **Objective**: Continuous locomotion on medium roughness
- **Configuration**:
  - `roughnessScale = 1.0`
  - `amplitude = 0.6`
  - `terrainSize = 10m`
- **Duration**: 5 minutes continuous movement
- **Success Criteria**: Maintain locomotion throughout without manual intervention

#### Scenario 3: Slope Variation

- **Objective**: Handle terrain with varying slopes
- **Configuration**: Multiple noise layers with different frequencies
- **Parameters**:
  - `useMultipleLayers = true`
  - `secondLayerStrength = 0.5`
  - `thirdLayerStrength = 0.25`
- **Success Criteria**: Adapt gait to terrain variations automatically

### C. Ramp Terrain Tests

#### Scenario 1: Steepness Challenge

- **Objective**: Climb ramps of increasing steepness
- **Configurations**:

| Height Difference | Approximate Angle | Difficulty |
|---|---|---|
| 0.5m | ~5° | Easy |
| 1.0m | ~10° | Medium |
| 1.5m | ~15° | Hard |
| 2.0m | ~20° | Extreme |

- **Fixed Parameters**:
  - `platformSize = 4m`
  - `rampWidth = 2m`
  - `rampLength = 10m`
- **Success Criteria**: Successfully climb to second platform

#### Scenario 2: Precision Approach

- **Objective**: Precise positioning on narrow ramp
- **Configurations**:
  - Wide ramp: `rampWidth = 2.0m`
  - Medium ramp: `rampWidth = 1.5m`
  - Narrow ramp: `rampWidth = 1.0m`
- **Fixed Parameters**: `heightDifference = 1.0m`
- **Success Criteria**: Stay on ramp surface without falling off sides

**Scenario 3: Bi-directional Test**

- **Objective**: Climb up and descend ramp successfully
- **Configuration**: Medium difficulty (`heightDifference = 1.0m`)
- **Success Criteria**: Complete round trip without falling
- **Measurement**: Time for complete cycle

---

# 3. Experimental Design

## Test Protocol

1. **Randomization**:
   - Randomize order of control methods per session
   - Randomize terrain configurations
   - Use Latin square design to control for order effects
2. **Repetitions**:
   - **N = 10** trials per scenario per control method
   - Total trials per terrain type: 60-90 trials
   - Total experiment: ~200 trials
3. **Baseline Establishment**:
   - Test both methods on flat terrain first
   - Establish baseline performance parameters
   - Calibrate measurement systems
4. **Environmental Controls**:
   - Same initial robot position and orientation
   - Identical environmental parameters
   - Reset robot state between trials
   - Same Unity physics settings

## Trial Procedure

1. **Pre-trial Setup** (30 seconds):
   - Reset hexapod to starting position
   - Initialize terrain configuration
   - Clear previous trial data
   - Verify all systems operational
2. **Trial Execution** (Variable duration):
   - Start data recording
   - Begin hexapod movement
   - Monitor for success/failure conditions

   - Record continuous performance data

3. **Post-trial Data Collection** (10 seconds):

   - Stop recording
   - Calculate derived metrics
   - Save trial data
   - Log any anomalies or observations

---

## 4. Specific Test Configurations

### Environment Parameters Summary

#### Obstacle Terrain

```
Terrain Configuration:
  area_size: [28.5m, 29m]
  floor_height: variable
  min_spacing: [0.3m, 0.5m, 0.8m]
  altura_extra: 0.1m

Obstacle Densities:
  low: 5 obstacles
  medium: 10 obstacles
  high: 15 obstacles

Test Variants: 9 configurations
```

#### Rough Terrain

```
Fixed Parameters:
  terrain_size: 10m
  terrain_resolution: 50
  enable_realtime_update: true

Variable Parameters:
  roughness_scale: [0.5, 1.0, 1.5, 2.0]
  amplitude: [0.3, 0.6, 1.0, 1.5]

Noise Layers:
  use_multiple_layers: true
  second_layer_strength: 0.5
  third_layer_strength: 0.25

Test Variants: 16 configurations
```

#### Ramp Terrain

```
Fixed Parameters:
  platform_size: 4m
  ramp_width: 2m
  ramp_length: 10m
  enable_realtime_update: true

Variable Parameters:
  height_difference: [0.5m, 1.0m, 1.5m, 2.0m]
  ramp_width_variants: [1.0m, 1.5m, 2.0m]

Test Variants: 12 configurations
```

---

## 5. Statistical Analysis Plan

### Comparative Analysis Methods

**Parametric Tests**

- **Two-sample t-tests**: Compare mean performance between IK and Neural methods
- **Paired t-tests**: Compare performance within subjects across conditions
- **ANOVA**: Multi-factor analysis (terrain type × control method × difficulty)
- **Repeated measures ANOVA**: Account for within-subject correlations

**Non-parametric Tests**

- **Mann-Whitney U test**: Compare medians when normality assumptions violated
- **Wilcoxon signed-rank test**: Paired non-parametric comparisons
- **Kruskal-Wallis test**: Multiple group comparisons

**Categorical Analysis**

- **Chi-square tests**: Success rate comparisons
- **Fisher's exact test**: Small sample categorical comparisons
- **McNemar's test**: Paired categorical data

### Effect Size Measurements

| Metric | Effect Size Measure | Interpretation |
|---|---|---|
| Continuous variables | Cohen's d | Small: 0.2, Medium: 0.5, Large: 0.8 |
| Success rates | Odds ratio | OR > 2 or < 0.5 considered meaningful |
| All metrics | Confidence intervals | 95% CI for practical significance |

### Sample Size Justification

- **Power analysis**: 80% power to detect medium effect sizes ($d = 0.5$)

- **Alpha level**: 0.05 (two-tailed tests)
- **Expected effect size**: Based on pilot testing
- **Minimum detectable difference**: 15% for continuous variables, 20% for success rates

---

## 6. Implementation Timeline

### Phase 1: Baseline Establishment (Week 1)

- [ ] Implement data collection system
- [ ] Test both control methods on flat terrain
- [ ] Validate measurement accuracy
- [ ] Establish baseline performance parameters
- [ ] Calibrate stability and energy metrics

### Phase 2: Obstacle Terrain Testing (Week 2)

- [ ] Execute 9 configurations × 2 methods × 10 trials = 180 trials
- [ ] Daily data backup and preliminary analysis
- [ ] Monitor for any systematic issues
- [ ] Adjust parameters if necessary

### Phase 3: Rough Terrain Testing (Week 3)

- [ ] Execute 16 configurations × 2 methods × 10 trials = 320 trials
- [ ] Focus on endurance and adaptability metrics
- [ ] Document terrain generation consistency
- [ ] Analyze real-time adaptation capabilities

### Phase 4: Ramp Terrain Testing (Week 4)

- [ ] Execute 12 configurations × 2 methods × 10 trials = 240 trials
- [ ] Emphasize precision and stability metrics
- [ ] Test bi-directional movement capabilities
- [ ] Validate angle calculations

### Phase 5: Analysis and Validation (Week 5)

- [ ] Complete statistical analysis
- [ ] Generate performance visualizations
- [ ] Identify significant differences
- [ ] Conduct targeted validation tests
- [ ] Prepare final report

---

## 7. Expected Outcomes and Hypotheses

### Primary Hypotheses

**H1: Terrain-Specific Performance**

- **Hypothesis**: Neural circuit will demonstrate superior performance on rough terrain due to adaptive capabilities
- **Prediction**: Neural > IK for rough terrain success rates and stability
- **Measurement**: Success rate difference > 20%, stability score difference > 0.5

#### H2: Precision Tasks

- **Hypothesis**: IK will show higher precision in obstacle navigation and narrow passage traversal
- **Prediction**: IK > Neural for precision metrics and path deviation
- **Measurement**: Path deviation difference > 0.2m, precision difference > 0.1m

#### H3: Energy Efficiency

- **Hypothesis**: Energy efficiency will vary significantly between methods and terrain types
- **Prediction**: Neural circuit more efficient on complex terrain, IK more efficient on simple terrain
- **Measurement**: Energy consumption difference > 15%

#### H4: Adaptability and Robustness

- **Hypothesis**: Neural circuit will exhibit faster adaptation to environmental changes
- **Prediction**: Neural < IK for adaptation time, Neural > IK for disturbance recovery
- **Measurement**: Adaptation time difference > 2 seconds

### Secondary Hypotheses

#### H5: Learning Effects

- **Hypothesis**: Neural circuit performance will improve over multiple trials
- **Prediction**: Significant positive correlation between trial number and performance for Neural method
- **Analysis**: Linear regression of performance vs. trial number

#### H6: Terrain Complexity Interaction

- **Hypothesis**: Performance difference between methods will increase with terrain complexity
- **Prediction**: Interaction effect between control method and difficulty level
- **Analysis**: Two-way ANOVA with interaction terms

### Decision Criteria

#### Statistical Significance

- **Primary analyses**: $p < 0.05$ (Bonferroni correction for multiple comparisons)
- **Exploratory analyses**: $p < 0.10$
- **Effect size threshold**: Cohen's d > 0.5 for practical significance

#### Practical Significance Thresholds

- **Success rate**: Difference > 20%
- **Completion time**: Difference > 15%

- **Energy efficiency**: Difference > 15%
- **Stability**: Difference > 0.5 standard deviations
- **Path deviation**: Difference > 0.2m

---

# 8. Data Management and Quality Control

## Data Collection Standards

### File Naming Convention

```
HexapodTest_[ControlMethod]_[TerrainType]_[Configuration]_[TrialNum]_[Timestamp].csv
Example: HexapodTest_IK_Obstacle_Medium_T005_20241201_143022.csv
```

### Data Validation Checks

- [ ] Verify all metrics within expected ranges
- [ ] Check for missing data points
- [ ] Validate timestamp consistency
- [ ] Confirm trial completion status
- [ ] Cross-reference environmental parameters

### Backup and Version Control

- [ ] Daily backup of all trial data
- [ ] Version control for analysis scripts
- [ ] Documentation of any parameter changes
- [ ] Maintain raw data integrity

## Quality Assurance

### Pre-trial Checks

- [ ] Robot systems functional
- [ ] Terrain properly configured
- [ ] Data logging systems active
- [ ] Environmental conditions stable

### During-trial Monitoring

- [ ] Real-time data validation
- [ ] System performance monitoring
- [ ] Anomaly detection and logging
- [ ] Manual observation notes

### Post-trial Verification

- [ ] Data completeness check
- [ ] Metric calculation verification
- [ ] Outlier identification
- [ ] Trial validity assessment

---