LEGO Robots with ZCS controllers in continuous world

Introduction

Zeroth level Classifier System

Model

Implementation
- Software
- Robot
- Worlds
- Parameter optimization

Experiments
- Hand-made controller
- Hypothesis
- Design
- Analysis

Conclusions
- Summary
- Contributions
- Future work

Questions

LEGO Mindstorms robots with a ZCS controller in a simulated physical world

2^{nd} KIM presentation

Ernst Blaauw

Faculteit der Exacte Wetenschappen
Supervisor: dr. M.C. Schut

Wednesday, the 13^{th} of January, 2010

vrije Universiteit *amsterdam*

Table of contents

1. Introduction
2. Zeroth level Classifier System
3. Model
4. Implementation
   - Software
   - Robot
   - Worlds
   - Parameter optimization
5. Experiments
   - Hand-made controller
   - Hypothesis
   - Design
   - Analysis
6. Conclusions

Please ask questions during talk!

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

What?

- A simulation in worlds with physics
- A ZCS controller runs on a robot
- The task is surveillance

Why?

- A ZCS controller has only been used in grid worlds
- ZCS controller consists of rules $\longrightarrow$ learn!
- Relatively simple lay-out

Objectives

Main goals of thesis

Objectives

1. Demonstrate feasibility of an existing ZCS controller on a LEGO Mindstorms robot in a surveillance task using a simulation
2. Show usefulness of a ZCS controller in a continuous world with continuous time

Research questions

- Advantages / disadvantages ZCS?
- ZCS in continuous world?
- Performance across worlds?
- Parameter optimization useful?

1 Introduction

2 Zeroth level Classifier System

3 Model

4 Implementation
- Software
- Robot
- Worlds
- Parameter optimization

5 Experiments
- Hand-made controller
- Hypothesis
- Design
- Analysis

6 Conclusions

Introduction

Zeroth level Classifier System

Model

Implementation
- Software
- Robot
- Worlds
- Parameter optimization

Experiments
- Hand-made controller
- Hypothesis
- Design
- Analysis

Conclusions
- Summary
- Contributions
- Future work

Questions

ZCS overview

General characteristics of a ZCS

ZCS is an evolutionary machine learning system.

- Published by Wilson (1994)
- Rule based
- Binary input / output
- Generalize

ZCS overview

General characteristics of a ZCS

ZCS is an evolutionary machine learning system.

- Published by Wilson (1994)
- Rule based
- Binary input / output
- Generalize

Rule format:

{input}	:	{action}	:	{strength}
1#0	:	001	:	20
00#	:	111	:	15

ZCS overview

General characteristics of a ZCS

ZCS is an evolutionary machine learning system.

- Published by Wilson (1994)
- Rule based
- Binary input / output
- Generalize

Rule format:

{input}	:	{action}	:	{strength}
1#0	:	001	:	20
00#	:	111	:	15

- Rules fired if *input* matches
- Higher *strength* $\rightarrow$higher prob. on *action*
- GA creates new rules
- ZCS generalizes using #'s

ZCS - performance cycle

From input to execution of action

ZCS algorithm: main loop (performance cycle)

1: initialize population
2: repeat until (termination condition is satisfied)
3: select matching rule set $\mathcal{M}$ from population
4: select action set $\mathcal{A}$ from $\mathcal{M}$
5: select action from $\mathcal{A}$
6: run GA with probability χ
7: evaluate $\mathcal{A}$ using feedback from environment
8: end repeat

ZCS - discovery cycle

Discovery of new rules

ZCS algorithm: Genetic Algorithm (GA) (discovery cycle)

1: select two parents using fitness proportionate selection
2: recombine parents to create two kids
3: mutate kids with probability μ per allele
4: delete two classifiers from population selected using inverse fitness proportionate selection

Example

Simplified example of ZCS rules

Input:	**Actions:**
00: cold and dry	0: don't wear jacket
01: cold and rain	1: wear jacket
10: hot and dry	
11: hot and rain	

Best solution: rain and/or cold $\rightarrow$ jacket
hot and dry $\rightarrow$ no jacket

Example

Simplified example of ZCS rules

Input:	**Actions:**
00: cold and dry	0: don't wear jacket
01: cold and rain	1: wear jacket
10: hot and dry	
11: hot and rain	

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Weather: cold and dry (00)

Rules:

00:0:20
00:1:15
01:0:20
11:1:30
11:0:25

Example

Simplified example of ZCS rules

Input:
00: cold and dry
01: cold and rain
10: hot and dry
11: hot and rain

Actions:
0: don't wear jacket
1: wear jacket

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Weather: cold and dry (00)

Rules:

00:0:20
00:1:15
01:0:20
11:1:30
11:0:25

Example

Simplified example of ZCS rules

Input:
00: cold and dry
01: cold and rain
10: hot and dry
11: hot and rain

Actions:
0: don't wear jacket
1: wear jacket

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Weather: cold and dry (00)
Selected rule:
00:0:20
Action: No jacket

Example

Simplified example of ZCS rules

Input:	**Actions:**
00: cold and dry	0: don't wear jacket
01: cold and rain	1: wear jacket
10: hot and dry	
11: hot and rain	

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Weather: cold and dry (00)

Bad action, rule penalized:

00:0:10

Example

Simplified example of ZCS rules

Input:	**Actions:**
00: cold and dry	0: don't wear jacket
01: cold and rain	1: wear jacket
10: hot and dry	
11: hot and rain	

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Weather: hot and rain (11)

Rules:

00:0:10
00:1:15
01:0:20
11:1:30
11:0:25

Example

Simplified example of ZCS rules

Input:	**Actions:**
00: cold and dry	0: don't wear jacket
01: cold and rain	1: wear jacket
10: hot and dry	
11: hot and rain	

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Weather: hot and rain (11)

Rules:

00:0:10
00:1:15
01:0:20
11:1:30
11:0:25

Example

Simplified example of ZCS rules

Input:	**Actions:**
00: cold and dry	0: don't wear jacket
01: cold and rain	1: wear jacket
10: hot and dry	
11: hot and rain	

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Weather: hot and rain (11)

Selected rule:

11:1:30

Action: Wear jacket

Example

Simplified example of ZCS rules

Input:	**Actions:**
00: cold and dry	0: don't wear jacket
01: cold and rain	1: wear jacket
10: hot and dry	
11: hot and rain	

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Weather: hot and rain (11)

Good action, rule rewarded:

11:1:40

Example

Simplified example of ZCS rules

Input:	**Actions:**
00: cold and dry	0: don't wear jacket
01: cold and rain	1: wear jacket
10: hot and dry	
11: hot and rain	

Best solution: rain and/or cold → jacket
hot and dry → no jacket

GA invoked, new rule:
10:0:20

Worst rule thrown away:
~~**00:0:10**~~

Example

Simplified example of ZCS rules

Input:	**Actions:**
00: cold and dry	0: don't wear jacket
01: cold and rain	1: wear jacket
10: hot and dry	
11: hot and rain	

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Weather: hot and dry (10)

Rules:

10:0:20
00:1:15
01:0:20
11:1:40
11:0:25

Example

Simplified example of ZCS rules

Input:	**Actions:**
00: cold and dry	0: don't wear jacket
01: cold and rain	1: wear jacket
10: hot and dry	
11: hot and rain	

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Weather: hot and dry (10)

Rules:

10:0:20

00:1:15

01:0:20

11:1:40

11:0:25

Example

Simplified example of ZCS rules

Input:	**Actions:**
00: cold and dry	0: don't wear jacket
01: cold and rain	1: wear jacket
10: hot and dry	
11: hot and rain	

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Weather: hot and dry (10)

Selected rule:

10:0:20

Action: No jacket

Example

Simplified example of ZCS rules

Input:
00: cold and dry
01: cold and rain
10: hot and dry
11: hot and rain

Actions:
0: don't wear jacket
1: wear jacket

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Weather: cold and dry (00)
Good action, rule rewarded:
10:0:30

Example

Simplified example of ZCS rules

Input:	**Actions:**
00: cold and dry	0: don't wear jacket
01: cold and rain	1: wear jacket
10: hot and dry	
11: hot and rain	

Best solution: rain and/or cold → jacket
hot and dry → no jacket

Rules:

10:0:30
00:1:15
01:0:20
11:1:40
11:0:25

1 Introduction

2 Zeroth level Classifier System

3 Model

4 Implementation
- Software
- Robot
- Worlds
- Parameter optimization

5 Experiments
- Hand-made controller
- Hypothesis
- Design
- Analysis

6 Conclusions

Input vector

Information available to ZCS

6 genes:

- 1 Touch sensor on/off
- 2 + 3 Speed wheels relative to each other
- 4 Camera detects pen track
- 5 Camera detects more/less pen track than previous time step
- 6 Moving forwards/backwards

Output vector

Defining the actions

Robot can perform actions:

Vector	Action
000	turn left
001	turn right
010	move forwards
011	move backwards
100	turn left in place
101	turn right in place
110	turn back left
111	turn back right

1 Introduction

2 Zeroth level Classifier System

3 Model

4 Implementation
- Software
- Robot
- Worlds
- Parameter optimization

5 Experiments
- Hand-made controller
- Hypothesis
- Design
- Analysis

6 Conclusions

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Webots

Simulation software

Extensive simulation packet:

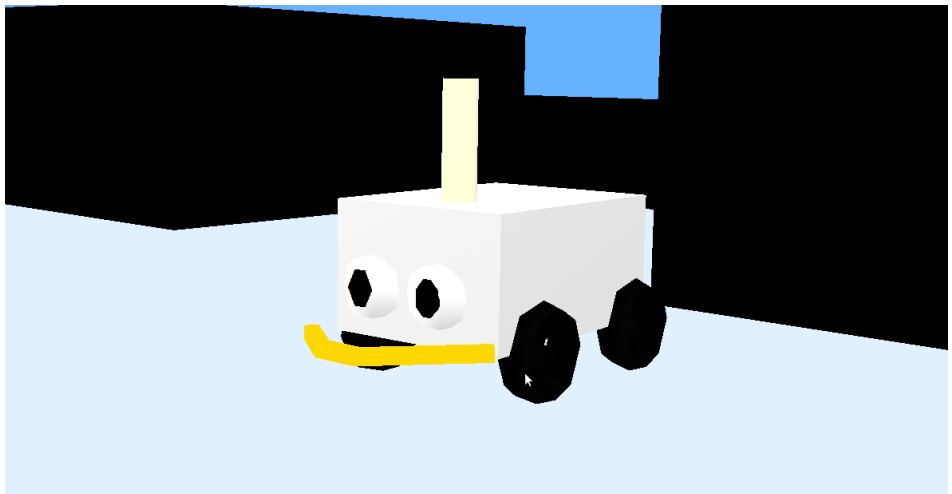
- Simulates physics
- A lot of sensors and actuators included
- Numerous examples
- Modelling in XML
- Controllers: C/C++, Java, Python

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Robot

LEGO Mindstorms robot

- Four wheels
- Touch sensor on the front side
- Camera device pointing to the ground
- Pen, which draws fading tracks on the ground

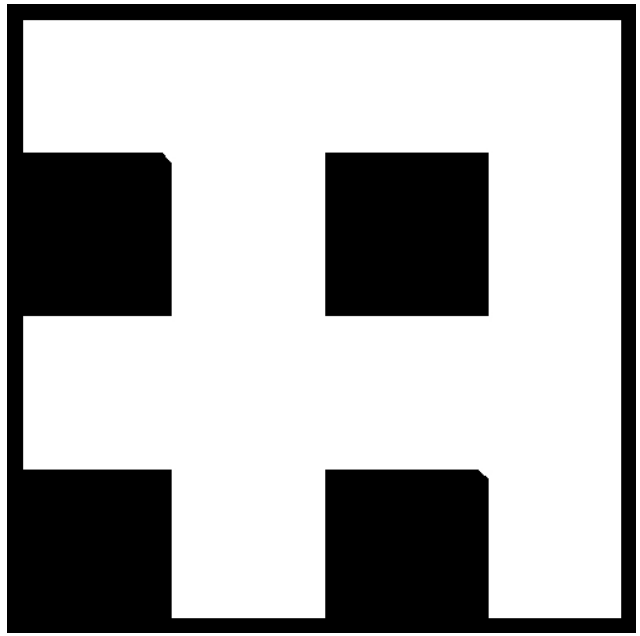


Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Woods1

.	.	.	.	.
.	.	.	.	.
t	t	f	.	.
t	t	t	.	.
t	t	t	.	.

Original world



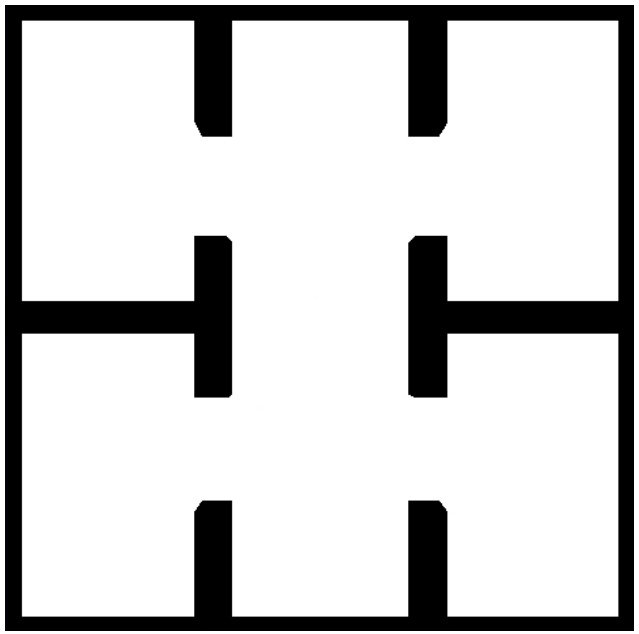
implementation in Webots (topview)

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Woods102

t	t	t	t	t	t	t
t	.	t	f	t	.	t
t	.	t	.	t	.	t
t	.	.	.	.	.	t
t	.	t	.	t	.	t
t	t	t	t	t	t	t
t	.	t	.	t	.	t
t	.	.	.	.	.	t
t	.	t	.	t	.	t
t	.	t	f	t	.	t
t	t	t	t	t	t	t

Original world

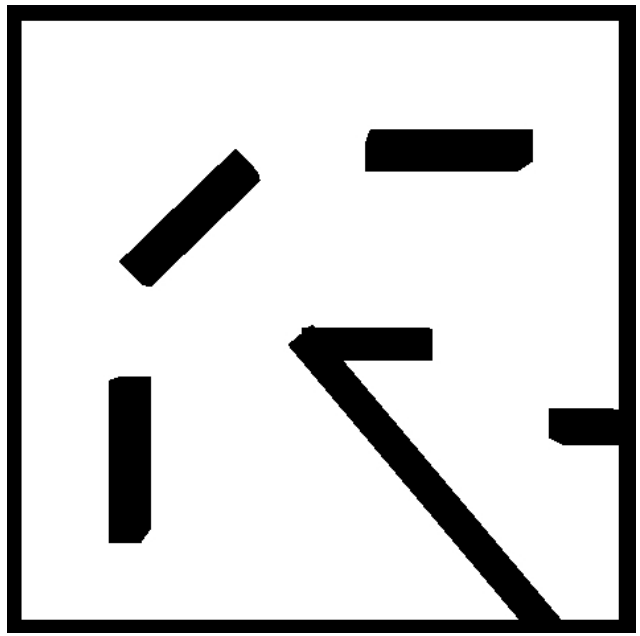


implementation in Webots (topview)

Maze5

t	t	t	t	t	t	t	t	t
t	.	.	.	.	.	.	f	t
t	.	.	t	.	t	t	.	t
t	.	t	.	.	.	.	.	t
t	.	.	.	t	t	.	.	t
t	.	t	.	t	.	.	t	t
t	.	t	.	.	t	.	.	t
t	.	.	.	.	.	t	.	t
t	t	t	t	t	t	t	t	t

Original world



implementation in Webots (topview)

REVAC

A meta-evolutionary optimizer to configure parameters

- Wilson parameters: best?
- REVAC automatically optimizes
- Using information theory to measure parameter relevance
- In grid worlds with ZCS: good results

1 Introduction

2 Zeroth level Classifier System

3 Model

4 Implementation
- Software
- Robot
- Worlds
- Parameter optimization

5 Experiments
- Hand-made controller
- Hypothesis
- Design
- Analysis

6 Conclusions

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Hand-made controller

Consisting of hand-crafted rules

The **hand-made controller** benchmarks ZCS controller.
Rules:

- Touch sensor: go back or left or right
- More pen track than earlier: go left or go right
- Less pen track than earlier: keep heading in direction
- Not on track: move forwards

Hypothesis

H_a *(main)*

The performance of the ZCS controller is at least as good as the performance of the hand-made controller

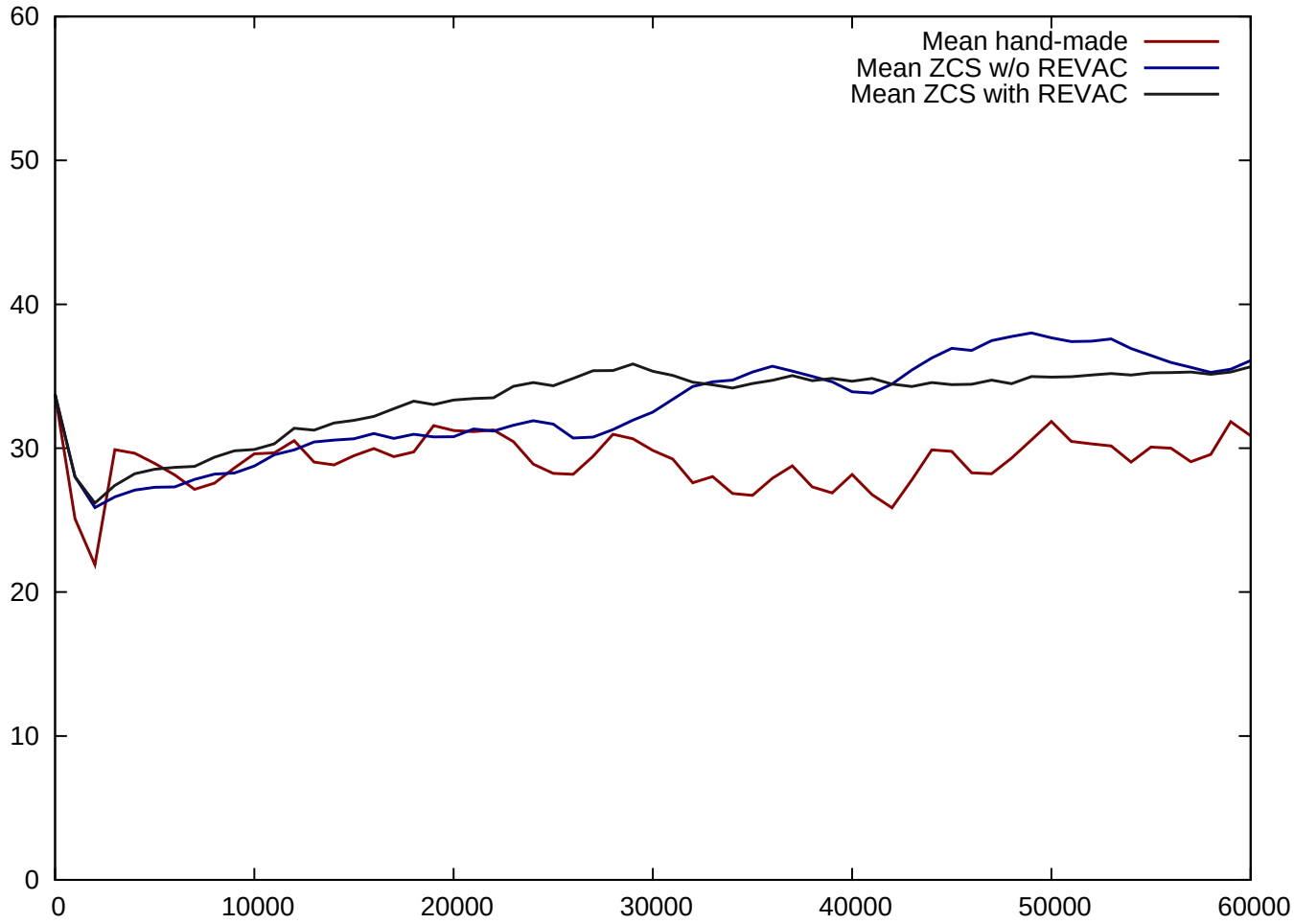
Design

- Independent: 2 controller
- Independent: 3 worlds
- Dependent: Distance to nearest pen track
- Each experiment contains three robots
- 60.000 time steps

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

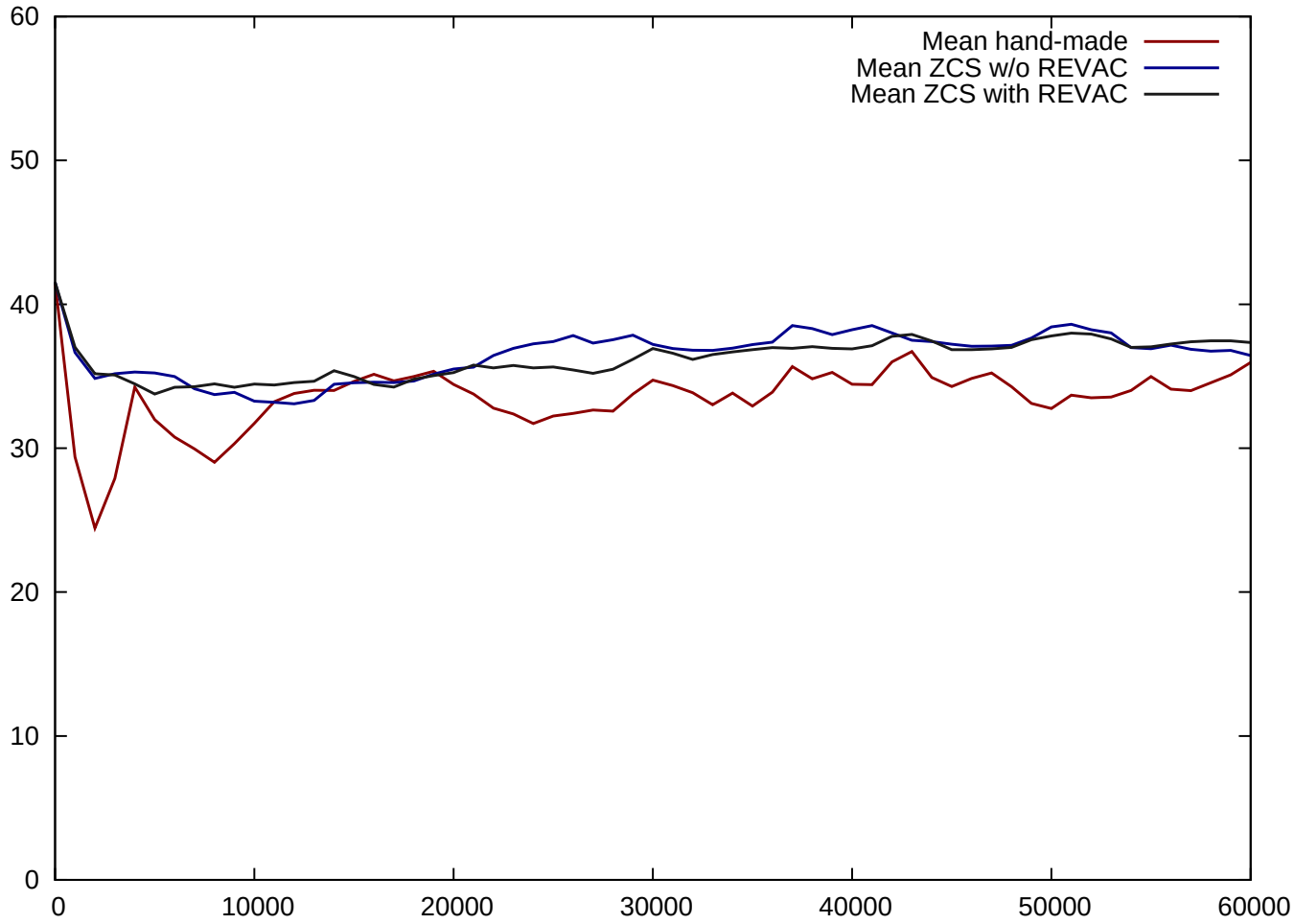
Analysis

Mean performance Woods1. Lower is better.



Analysis

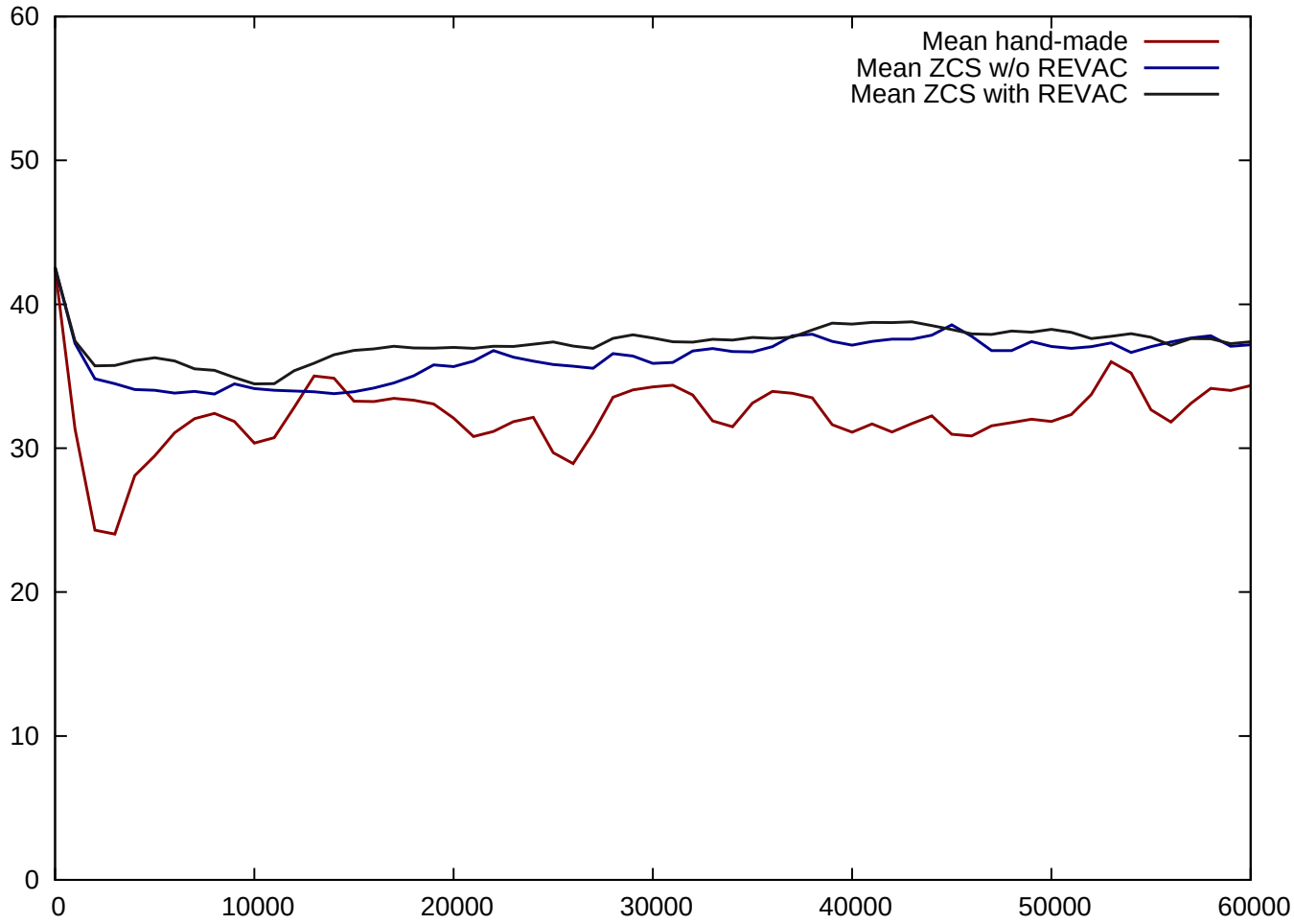
Mean performance Woods102. Lower is better.



Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Analysis

Mean performance Maze5. Lower is better.



Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Analysis

Summary of performance

- Hand-made controller performs better
- But has higher σ
- No difference between ZCS's

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Analysis

Summary of performance

- Hand-made controller performs better
- But has higher σ
- No difference between ZCS's
- Wilcoxon: hand-made controller performs better than ZCS
- Wilcoxon: no difference between ZCS's

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

1 Introduction

2 Zeroth level Classifier System

3 Model

4 Implementation
- Software
- Robot
- Worlds
- Parameter optimization

5 Experiments
- Hand-made controller
- Hypothesis
- Design
- Analysis

6 Conclusions

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Objectives

Main goals of thesis

Objectives

1. Demonstrate feasibility of an existing ZCS controller on a LEGO Mindstorms robot in a surveillance task using a simulation
2. Show usefulness of a ZCS controller in a continuous world with continuous time

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Research questions

- Advantages / disadvantages ZCS?
- ZCS in continuous world?
- Performance across worlds?
- Parameter optimization useful?

Hypothesis

Possibility to reject?

□ $H_{a\ (main)}$

The performance of the ZCS controller is at least as good as the performance of the hand-made controller

Hypothesis

Possibility to reject?

✗ $H_{a\ (main)}$

The performance of the ZCS controller is at least as good as the performance of the hand-made controller

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Contributions

- Software framework
- Research framework

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Future work

What's next?

- Memory
- Communication
- Real robots
- ...

Introduction
Zeroth level Classifier System
Model
Implementation
Software
Robot
Worlds
Parameter optimization
Experiments
Hand-made controller
Hypothesis
Design
Analysis
Conclusions
Summary
Contributions
Future work
Questions

Questions

Questions?

LEGO Mindstorms robots with a ZCS controller in a simulated physical world

Remarks?