Virtual Evolution of 2D Soft Robots

Naudé Conradie
Supervisor: Dr MP Venter

Department of Mechanical and Mechatronic Engineering, Stellenbosch University

22 November 2019

Overview

- Project scope

Overview

- Project scope
- Background

Overview

- Project scope
- Background
- Methodology

Overview

- Project scope
- Background
- Methodology
- Results and Conclusions

Project Scope

- Automate design of shape-changing soft robots

Project Scope

- Automate design of shape-changing soft robots
  - Change internal pressure

Project Scope

- Automate design of shape-changing soft robots
  - Change internal pressure
- Non-linear FEM

Project Scope

- Automate design of shape-changing soft robots
  - Change internal pressure
- Non-linear FEM
  - Restricted to two dimensions

Project Scope

- Automate design of shape-changing soft robots
  - Change internal pressure
- Non-linear FEM
  - Restricted to two dimensions
  - Modelled with real material properties

Project Scope (cont.)

- Computationally efficient

Project Scope (cont.)

- Computationally efficient
  - Use recursive grammatical encodings

Project Scope (cont.)

- Computationally efficient
  - Use recursive grammatical encodings
  - L-systems for cellular level

Project Scope (cont.)

- Computationally efficient
  - Use recursive grammatical encodings
  - L-systems for cellular level
  - CPPNs for organism level

Project Scope (cont.)

- Computationally efficient
  - Use recursive grammatical encodings
  - L-systems for cellular level
  - CPPNs for organism level
- Evolve a population to obtain best model

Background

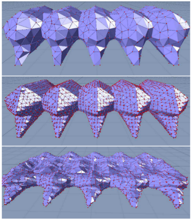
- Soft robotics

Background

- Soft robotics
  - Modelling and evolving soft bodies is computationally expensive

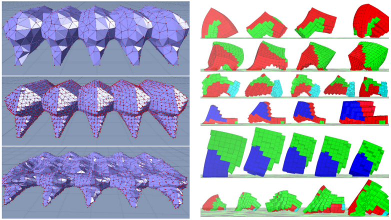
Background

- Soft robotics
  - Modelling and evolving soft bodies is computationally expensive

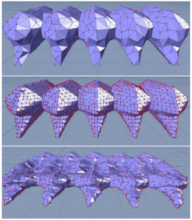
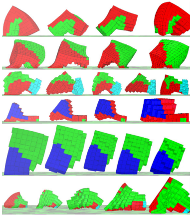
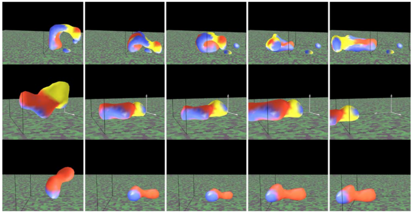


Background

- Soft robotics
  - Modelling and evolving soft bodies is computationally expensive



- Soft robotics
  - Modelling and evolving soft bodies is computationally expensive

Background (cont.)

- Lindenmayer systems (L-systems)

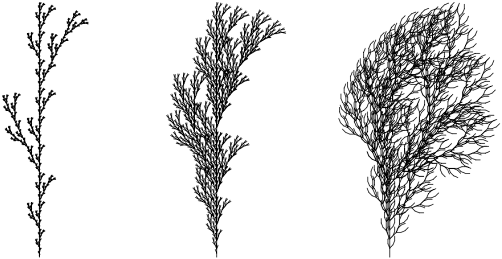
Background (cont.)

- Lindenmayer systems (L-systems)
  - Recursive grammatical encodings

Background (cont.)

- Lindenmayer systems (L-systems)
  - Recursive grammatical encodings
  - Built from axiom, variables, constants and rules

- Lindenmayer systems (L-systems)
  - Recursive grammatical encodings
  - Built from axiom, variables, constants and rules



a
n=5, δ=25.7°
F
F → F[+F]F[-F]F

b
n=5, δ=20°
F
F → F[+F]F[-F][F]

c
n=4, δ=22.5°
F
F → FF-[-F+F+F]+
[+F-F-F]

Background (cont.)

- Compositional Pattern-Producing Network - NeuroEvolution of Augmenting Technologies (CPPN-NEAT)

Background (cont.)

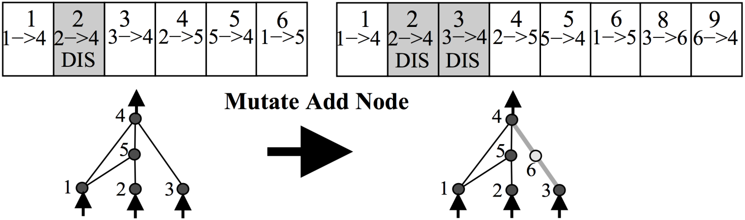
- Compositional Pattern-Producing Network - NeuroEvolution of Augmenting Technologies (CPPN-NEAT)
  - Neural networks

Background (cont.)

- Compositional Pattern-Producing Network - NeuroEvolution of Augmenting Technologies (CPPN-NEAT)
  - Neural networks
  - Evolved with topology augmentation

Background (cont.)

- Compositional Pattern-Producing Network - NeuroEvolution of Augmenting Technologies (CPPN-NEAT)
  - Neural networks
  - Evolved with topology augmentation



Software

- LSDyna

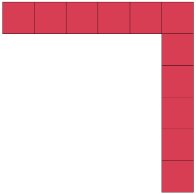
Software

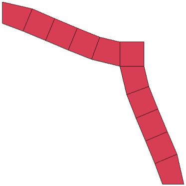
- LSDyna
  - Commercial software
  - Support

Software

- LSDyna
  - Commercial software
  - Support
  - High level of control
  - Robust

- LSDyna
  - Commercial software
  - Support
  - High level of control
  - Robust





Basic Structure

- Unit cell

Basic Structure

- Unit cell
  - Square

Basic Structure

- Unit cell
  - Square
  - Modelled with Mold Star 15

Basic Structure

- Unit cell
  - Square
  - Modelled with Mold Star 15
  - Predefined behaviours

Basic Structure

- Unit cell
  - Square
  - Modelled with Mold Star 15
  - Predefined behaviours

- Complete soft body
  - Constructed from unit cells

Basic Structure

- Unit cell
  - Square
  - Modelled with Mold Star 15
  - Predefined behaviours

- Complete soft body
  - Constructed from unit cells
  - Recursive grammatical encodings

Recursive Encodings

- L-systems

Recursive Encodings

- L-systems
  - Refer to unit cells

Recursive Encodings

- L-systems
  - Refer to unit cells
  - Construct soft body

Recursive Encodings

- L-systems
  - Refer to unit cells
  - Construct soft body
  - Genotype

Recursive Encodings

- L-systems
  - Refer to unit cells
  - Construct soft body
  - Genotype
- CPPN-NEAT

Recursive Encodings

- L-systems
  - Refer to unit cells
  - Construct soft body
  - Genotype
- CPPN-NEAT
  - Refer to whole body

Recursive Encodings

- L-systems
  - Refer to unit cells
  - Construct soft body
  - Genotype
- CPPN-NEAT
  - Refer to whole body
  - Phenotype

Proof of Concept

- Use material properties obtained from testing

Proof of Concept

- Use material properties obtained from testing
- Manufacture physical model

Proof of Concept

- Use material properties obtained from testing
- Manufacture physical model
  - Unit cell and whole body

Proof of Concept

- Use material properties obtained from testing
- Manufacture physical model
  - Unit cell and whole body
  - Produce at some thickness

Proof of Concept

- Use material properties obtained from testing
- Manufacture physical model
  - Unit cell and whole body
  - Produce at some thickness
  - Place between glass plates

Proof of Concept

- Use material properties obtained from testing
- Manufacture physical model
  - Unit cell and whole body
  - Produce at some thickness
  - Place between glass plates
  - Apply internal pressure

Proof of Concept

- Use material properties obtained from testing
- Manufacture physical model
  - Unit cell and whole body
  - Produce at some thickness
  - Place between glass plates
  - Apply internal pressure
  - Observe behaviour

Results and Conclusions

- Improve computing time required

Results and Conclusions

- Improve computing time required
- Prove practicality of recursive encodings

Results and Conclusions

- Improve computing time required
- Prove practicality of recursive encodings
- Replicable

Results and Conclusions

- Improve computing time required
- Prove practicality of recursive encodings
- Replicable
- Adaptable

Results and Conclusions

- Improve computing time required
- Prove practicality of recursive encodings
- Replicable
- Adaptable
  - 3D

Results and Conclusions

- Improve computing time required
- Prove practicality of recursive encodings
- Replicable
- Adaptable
  - 3D
  - Different objective functions

Questions?