SEMESTER PROJECT PRESENTATION:

ONLINE OPTIMIZATION OF
LOCOMOTION CONTROLLER FOR
ROOMBOTS

Student: Nguyen The Anh
Supervisors:
Rico Möckel, Stéphane Bonardi, Massimo Vespignani, Soha Pouya
Professor: Auke Jan Ijspeert

BIOROBOTICS LABORATORY
June 15, 2012
Outline

- Motivation
- Introduction
- Implementation
- Experiments
- Results
- Conclusion
Motivation

- Online learning of **optimal locomotion pattern**
- Adaptation to arbitrary structures and environmental conditions

I know how to move
I learn by myself

Previous works

• Online learning using CPG and Powell optimization method on YAMOR by A. Sproewitz et. al [1]
• Offline optimization using CPG and Particle Swarm Optimization (PSO) on Roombots in simulation by S. Pouya et. al [2]
• First steps towards online learning using CPG and PSO on Roombots without tracking system by F. Wilhelm [3]

➢ This project:

Online learning using CPG and PSO on Roombots with Kinect tracking system and GUI experiment software.
Roombots

- Modular self-reconfigurable robots
  - Designed for adaptive furniture
- A module:
  - 3 degrees of freedom (DOF)
  - 2 types of movement: Oscillation and Rotation
- Meta-module:
  - Two modules connected
  - 6 degrees of freedom (DOFs)
  - 4 configurations: PAR, PER, SRS, SRZ
Locomotion control and optimization

- Controlled by **Central Pattern Generator (CPG)**
  - Network of coupled phase oscillators
  - 1 oscillator per 1 degree of freedom
- A meta-module with 6 DOFs
  - Only PER configuration used
  - 6 DOFs corresponds to 6 oscillators
- One oscillator can generate:
  - Oscillation
  - Rotation
  - Locked
- Particle Swarm Optimization (**PSO**)  
  - Stochastic, population-based optimization method based on collaboration
  - Robust against local minima
  - Optimize the Euclidean distance between initial and final position of Roombots after travelling in 30s
Experimental environment

- **Experimental Setup with Kinect**
- **3 types of surface material:**
  - Cork mat (high friction)
  - Paper (medium friction)
  - Plastic rubber (low friction)
- **Software**
  - Calculate fitness values
  - Tracking System
  - Control Roombots
    - Commands via Bluetooth
- **Roombots**
  - CPG, PSO
Experiment software

- **GUI**
- **3 main control sections:**
  - **Tracking**
  - **Communication with Roombots**
  - **Control of optimization experiment**
- **Other sections:** Visual monitor, Software status, Bluetooth Data log
- **Convenient tool used for conducting experiments.**
Tracking & location detection algorithm

Background Image → Gaussian Blur Filter → Difference Image → Manual/Automatic Threshold → Threshold Diff. Image → Find Contours → Robot Position (x,y) → OpenCV Library

Current Image → Gaussian Blur Filter → Difference Image

Position estimation using central of mass:
- : center of rectangle
+ : center of mass
Optimization and control parameters

- Fitness: Distance travelled in 30s (average over 3 trials), zero if collision
- The 3rd Experiments:
- Reduction of number of CPG parameters:
  - 1 and 6 has little impact:
    - $R_1 = R_6 = X_1 = X_6 = 0$
  - Then, 2 and 5 become axial rotation invariant
    - $X_2 = X_5 = 0$
  - Assume symmetric amplitudes of 2,3,4,5
    - $R_2 = R_3 = R_4 = R_5 = R$
- 6 CPG parameters
  1. Amplitude $R$
  2. Offset $X_3$
  3. Offset $X_4$
  4. Coupling phase $\phi_{23}$
  5. Coupling phase $\phi_{34}$
  6. Coupling phase $\phi_{45}$
- The parameters and their ranges are set via XML file.
  - Software reads XML file and sends automatically commands to Roombots for settings.

In the 3rd experiment, the best gait was found.
Results

Parameters values vs. iterations of the best gait

Fitness values vs. iterations of the best gait

Fitness values of particles along iterations
Results

• A good gait: fitness value of 324 in average or 121.5cm (i.e. a speed of 4.05cm/s).

\[ R_2 = R_3 = R_4 = R_5 = 1.448, \quad X_3 = 0.708, \quad X_4 = 0.044, \quad \phi_{23} = 1.375, \quad \phi_{34} = 3.102, \quad \phi_{45} = 0.057 \]
Results

• The current best gait: fitness value of 395 in average or 150cm (i.e. a speed of 5cm/s).

\[ R_2 = R_3 = R_4 = R_5 = 2.065, X_3 = 0.407, X_4 = -0.03, \phi_{23} = 2.418, \phi_{34} = 3.103, \phi_{45} = 0 \]
Gafts evaluations

- Gait vs. Friction
  - 3 materials, same initial state:

  ![Materials](Cork mat, Paper, Plastic rubber)

  - Different frictions affect the performance of a gait
    - The difference depends on how much the gait uses friction to move
Gaits evaluations

- Gait vs. Initial states
  - 4 initial states, 2 materials:
    - Different initial states can ruin the performance of a gait
      - Because the gait was learnt from a certain initial state.
      - Due to the mechanical symmetry in Roombots the difference of the gait performances within two pairs: side 1 & side 2, and side 3 & side 4 is small.
Gaits evaluations

- Gait vs. Obstacles
  - Gait was learnt in the condition where there is no obstacle but a learnt gaits can be still robust against obstacle.
  - The distance to an obstacle affects the performance of a gait
  - The friction can improve the robustness of gait over obstacle

- The best gait is robust against obstacles.

Good gait vs. obstacle

Best gait vs. obstacle
Gaits evaluations

- Gait vs. Slope
  - Gait was learnt in the condition where the surface is flat but it still works when there is a slope.
  - A high friction can improve the performance of gait over slope
  - Roombots tends to fall down the slope when the surface is slippy
    - The good gait can reach the top, but the best gait fell down the slope
Conclusion

- Implemented an **efficient tracking system** for locomotion online learning experiment.
- The **user-friendly software** with the tracking system makes the experiment more convenient, time-saving, and energy-saving.
  - Fully supports loading experiment settings, setting CPG parameters for a gait, setting CPG parameters’ ranges
  - Load and save PSO particles from file to Roombots and reversely.
- Debugged critical bugs in CPG firmware code in Roombots.
- Added many commands and a feature allowing users to **select CPG parameters** used in PSO **without modifying firmware**.
- Found two interesting gaits.
- Various **gait evaluations** were conducted.
  - Initial state and surface friction are two main factors that affect to the performance of a particular gait.
Future works

- PSO with velocity (vector) rather than speed (scalar).
- Ability to return the initial state and position automatically based on vision data
- Use internal sensors such as accelerometers or gyroscopes to compute the fitness value
Thank you for your attention

QUESTIONS???
References


Experiment procedures

1. Load experiment Settings
2. Load PSO particles (resume exp) or Init PSO particles (new exp)
3. Run CPG simulation to detect collision
   1. If there is a collision, f = 0, go to 5
   2. If there is no collision, go to 4
4. Run CPG controller-based for 30s, \textit{fitness value}(f) = travelled distance
5. Set the fitness value of current particle and Go to next particles
6. Repeat step 3 until we want to pause or terminate experiments
7. Save PSO particles when pause the experiment