Abstract
Locomotion has been a long-standing research issue in a number of scientific communities including biology, biomechanics, neuroscience, artificial intelligence, and robotics. While there are many different cross-sections to understand the problem of locomotion, we will focus on this fascinating problem from a viewpoint of embodied artificial intelligence in this lecture.
The goal of this lecture series is threefold. First, by looking into previous and current locomotion research around the world, we will obtain the basic motivation of locomotion studies. The problems explored in locomotion research are common in surprisingly many disciplines, and applicable to a number of similar problems. Second, we will learn useful models and tools, not only for the beginners of the field, but also for a systematic exploration of the complex problem. Owing to its relatively long history of research, locomotion research has developed a number of tools for systematic understanding of complex behavior of animals and machines, which are also useful for many different complex problems. And third, in order to become familiar with the concepts and tools, some hands-on exercises will be offered.

Short Biography of the Lecturer:
Fumiya Iida received his bachelor and master degrees in mechanical engineering at Tokyo University of Science (Japan), and Ph.D. at the Faculty of Science, University of Zurich (Switzerland, 2006). From 2004 to 2005, he was also engaged in biomechanics research of human locomotion at Locomotion Laboratory, University of Jena (Germany). He is currently a postdoctoral associate at the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology. He has been involved in a number of research projects related to robotics, embodied artificial intelligence and biomechanics, and his research interest includes biologically inspired robotics, navigation of autonomous robots, and dynamic legged locomotion. He serves as the editorial board of a few journals and the program committee member of a dozen of international conferences, and he was a recipient of the Prospective Researcher Fellowship and Professorship of the Swiss National Science Foundation. From August 2009, he starts up a new research group, Bio-Inspired Robotics Laboratory, at the Swiss Federal Institute of Technology (ETH Zurich).
Schedule “Dynamic Locomotion”

08:15-09:00 Lecture 1: Embodied AI and Dynamic Locomotion
   1.1 Goals of Embodied AI
   1.2 Embodied AI and Motor Control
   1.3 Embodied AI and Dynamic Locomotion
   1.4 Overview of Locomotion Studies
09:00-0915: Coffee Break

09:15-10:45 Lecture 2: Dynamics: Modeling and Analytical Tools
   2.1 What is Dynamics?
   2.2 General Framework of Modeling Dynamics
   2.3 Example: Pendulum Dynamics
   2.4 Example: Simulation of Walking
   2.5 Analysis of Stability
   2.6 Other dynamic models
10:45-11:00: Coffee Break

11:00-12:15 Lecture 3: Control & Optimization
   3.1 Motor Control in Nature and Robot
   3.2 Model Based Control and Optimization
   3.3 Model Free Control and Optimization: Reinforcement Learning
   3.4 Examples: Locomotion Control in Rough Terrain
   3.5 Other Control and Optimization

12:15-13:00 Lecture 4: Exercise
   4.1 Goals of Exercise
   4.2 Simulation Platform
   4.3 Task Description

13:00-17:00 Lunch, Exercise, Break

17:00-18:00 Lecture 5: Summary and Outlook
   5.1 Summary of Exercise
   5.2 Discussion: Trends and Challenges
   5.3 Conclusions
Exercise: Simulation of Compass Gait Model

Goals of Exercise
By using the simulation code of the compass gait model, you learn how physical dynamics are related to behaviors of bipedal walking. Through the assignments listed below, in particular, you will be able to understand how to design self-stability of a passive dynamic walker, how to analyze stability with visualization tools, and how to design controller and learning architectures for dynamic robots. Have fun!

Brief Description of the Compass Gait Model
As briefly introduced in the lectures, the compass gait model simulation consists of Continuous Swing Leg Dynamics and Collision Dynamics. In this exercise, you will have a complete sample code of compass gait model with a sinusoidal controller, in which you can easily manipulate various morphological and control parameters to explore variations of behaviors. For more detailed information, refer to Tutorial.

What you do in the exercise. *The tasks with * below are optional.

(0) Start MainStcipt.m and see whether the simulation works.
   (If not, see Tutorial.)

(1) Change initial conditions (th_0 th_1, th_dot_0, th_dot_1)
    morphology parameters (m, mH, a, b)
    control parameters (AMP, PER)
    and observe walking stability by the use of phase plots.

(2) Create passive dynamic walker.

(3) Change environment parameters
    switch between env00 (flat), env01(uhill), env02 (downhill),
    env11 (one step), and evn12 (random)
    and observe whether it can walk in different environments.

(4) Create envXX File and test it.
(5)* Design feedback controller (i.e. change Controller.m)
Try to design $h_a=f(q, q_d)$ such that the model can maintain stability in rough terrains. Would a PD controller (e.g. $h_a=K_p(q^*-q)+K_d(q_d^*-q_d)$) work in rough terrains?

(6)* Create your own physics model (i.e. changing continuous and collision dynamics)

(7)* Design your own learning architectures.

**Walking Competition**
If you manage to design a good set of morphological and control parameters of the compass gait model, we will do a competition. At the end of the exercise, you submit your parameter files to the Lecturer, and see which simulation works the best with respect to:

**Competition Metrics**

**Stability**
- First we test basic stability in the flat terrain
- Then add one-shot noise to see whether it recovers from it

**Adaptability**
- Test if your model can walk through a known rough terrain
- Then compute speed and energy efficiency
- Test if your model can be adaptive against some noise
- Test if it can adapt to unknown terrains

**Innovation**
- Develop your own physics model (continuous and collision)
- Develop your own controller (feedback, CPGs, etc.)
- Learning architecture?
Tutorial for the Compass Gait Simulator

There is an accompanying web site for the Dynamic Locomotion lecture (see Fig. 0)

http://people.csail.mit.edu/g_gomez/compassSimulator/

This website offers the following information

- Brief Introduction of the lecture
- Short biography of the lecturer (Dr. Fumiya Iida)
- Schedule
- A tutorial for the exercise (This document)
- The exercise
- A pointer to general MATLAB documentation
- HTML documentation of the source code provided for the exercise
- A pointer to download the latest version of the source code for the exercise
- Code structure and pointers to download individual files
- Details about the competition
- Reference materials for the lecture
Getting started

- Download the source code for the compass gait simulator from the following URL:
  http://people.csail.mit.edu/g_gomez/compassSimulator/code/compassGaitSimulator.tar

- Uncompress the source code in some directory (e.g., /home/matlabUser/) issuing the command:

  ```
  tar -xvf compassGaitSimulator.tar
  ```

This will uncompress the files for the compass gait simulator in a directory with the following structure:

```
/home/matlabUser/compassGaitSimulator
  documentation/
    lidaTedrakeAR4.sub.pdf
    Exercise
    Tutorial (this file)
  env/
    env00.mat
    env01.mat
    env02.mat
  CollisionModel.m
  ContinuousDynamics.m
  Controller.m
  ControlParam.m
  generateTerrain.fig
  generateTerrain.m
  InitCondition.m
  MainScript.m
  maxWindow.m
  MorphologyParam.m
```
Source code Structure

Main Script  
(MainScript.m)

Parameter Files

Morphology  
(MorphologyParam.m)

Control  
(ControlParam.m)

Environment  
(EnvGene .m, generateTerrain.m, generateTerrain.fig)

Functions that implement the Compass Gait Model

Initial conditions  
(InitCondition.m)

Continuous Swing Leg Dynamics  
(ContinuousDynamics.m)

Collision Model  
(CollisionModel.m)

Controller  
(Controller.m)

Utilities

Visualization

Maximize figure on screen  
(maxWindow.m)

Environment files

Flat terrain  
(env00.mat)

One small step  
(env01.mat)

Random Terrain  
(env02.mat)
Set the MATLAB current directory to point to the directory where you have the source code for the Compass Gait Simulator as illustrated in Figs 1 and 2.

Fig. 1

Fig. 2
In order to execute the Compass Gait Simulator, go to the MATLAB command window and issue the command “MainScript,” the compass gait simulator will start and a Figure will appear as illustrated in Fig. 3.
How to generate the terrain files

Terrain files can be found under the directory “env”. At the moment the terrain files have the following names

- env00.mat  Flat terrain
- env01.mat  Terrain with one small step
- env02.mat  Rough terrain

There are two ways to generate the terrain files:

- **EnvGene.m**
  Start the script “EnvGene” from the command window. This will generate all the terrain files in the directory “env”

- **generateTerrain.m**
  Start the Graphical User Interface called “generateTerrain” from the command window. A GUI will appear where you can select one or more types of terrain to generate as shown in Figs. 4 and 5.

Fig. 4
Click on the button “Generate terrain.” The file(s) will be generated under the “env” directory.

Note:

This method might require a newer version of MATLAB. It was only tested in MATLAB 7.8.0 R2009a.
Loading terrain files

Open the file MainScript.m and look for the following comments:

%%%% Set Parameters %%%%

% envxx.mat files are generated by running generateTerrain.m

And uncomment one of these lines to load the corresponding terrain file

% envFileName = ['env' filesep 'env00']; % flat ground
% envFileName = ['env' filesep 'env01']; % uphill
% envFileName = ['env' filesep 'env02']; % downhill
% envFileName = ['env' filesep 'env11']; % one small step
envFileName = ['env' filesep 'env12']; % randomly generated rough terrain

The use of “filesep” ensures that the directory path is platform independent (i.e., env\env00.mat in Windows, env/env00.mat in Linux)
Browsing the online documentation

The online documentation can be easily browsed online at the following URL:

http://people.csail.mit.edu/g_gomez/compassSimulator/doc/index.html

Fig. 6 shows an example of browsing the online documentation for the “MainScript.m”, when the user points to the “ContinuousDynamics” function, a tooltip appears giving the prototype of the function, by clicking on the link the browser will show the documentation for the “ContinuousDynamics” function.